Red Hat Linux 8.0

The Official Red Hat Linux System Administration Primer

Red Hat Linux 8.0: The Official Red Hat Linux System Administration Primer


Red Hat, Inc.

1801 Varsity Drive
Raleigh NC 27606-2072 USA
Phone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701
PO Box 13588
Research Triangle Park NC 27709 USA

rhl-sap(EN)-8.0-Print-RHI (2002-10-01T17:12-0400)

The GPG fingerprint of the security@redhat.com key is:
CA 20 86 86 2B D6 9D FC 65 F6 EC C4 21 91 80 CD DB 42 A6 0E

Table of Contents

Introduction

Welcome to the *Official Red Hat Linux System Administration Primer*.

The *Official Red Hat Linux System Administration Primer* contains introductory information for new Red Hat Linux system administrators. It will *not* teach you how to perform a particular task; rather, it will provide you with the background knowledge that more experienced system administrators have learned over time.

This guide assumes you have a limited amount of experience as a Linux user, but no Linux system administration experience. If you are completely new to Linux in general (and Red Hat Linux in particular), you should start by reading the *Official Red Hat Linux Getting Started Guide*.

More experienced system administrators should skim the *Official Red Hat Linux System Administration Primer* for overall concepts, and then concentrate on using the *Official Red Hat Linux Customization Guide* for assistance in performing specific tasks in a Red Hat Linux environment. Administrators requiring more in-depth, factual information should refer to the *Official Red Hat Linux Reference Guide*.

Note

Although this manual reflects the most current information possible, you should read the *Red Hat Linux Release Notes* for information that may not have been available prior to our documentation being finalized. They can be found on the Red Hat Linux CD #1 and online at:

`http://www.redhat.com/docs/manuals/linux`

1. Document Conventions

When you read this manual, you will see that certain words are represented in different fonts, typefaces, sizes, and weights. This highlighting is systematic; different words are represented in the same style to indicate their inclusion in a specific category. The types of words that are represented this way include the following:

`command`

Linux commands (and other operating system commands, when used) are represented this way. This style should indicate to you that you can type the word or phrase on the command line and press [Enter] to invoke a command. Sometimes a command contains words that would be displayed in a different style on their own (such as filenames). In these cases, they are considered to be part of the command, so the entire phrase will be displayed as a command. For example:

Use the `cat testfile` command to view the contents of a file, named `testfile`, in the current working directory.

`filename`

Filenames, directory names, paths, and RPM package names are represented this way. This style should indicate that a particular file or directory exists by that name on your Red Hat Linux system. Examples:

The `.bashrc` file in your home directory contains bash shell definitions and aliases for your own use.

The `/etc/fstab` file contains information about different system devices and filesystems.

Install the `webalizer` RPM if you want to use a Web server log file analysis program.

application

This style should indicate to you that the program named is an end-user application (as opposed to system software). For example:

Use **Mozilla** to browse the Web.

[key]

A key on the keyboard is shown in this style. For example:

To use [Tab] completion, type in a character and then press the [Tab] key. Your terminal will display the list of files in the directory that start with that letter.

[key]-[combination]

A combination of keystrokes is represented in this way. For example:

The [Ctrl]-[Alt]-[Backspace] key combination will exit your graphical session and return you to the graphical login screen or the console.

text found on a GUI interface

A title, word, or phrase found on a GUI interface screen or window will be shown in this style. When you see text shown in this style, it is being used to identify a particular GUI screen or an element on a GUI screen (such as text associated with a checkbox or field). Example:

Select the **Require Password** checkbox if you would like your screensaver to require a password before stopping.

top level of a menu on a GUI screen or window

When you see a word in this style, it indicates that the word is the top level of a pulldown menu. If you click on the word on the GUI screen, the rest of the menu should appear. For example:

Under **File** on a GNOME terminal, you will see the **New Tab** option that allows you to open multiple shell prompts in the same window.

If you need to type in a sequence of commands from a GUI menu, they will be shown like the following example:

Go to **Main Menu Button** (on the Panel) => **Programming** => **Emacs** to start the **Emacs** text editor.

button on a GUI screen or window

This style indicates that the text will be found on a clickable button on a GUI screen. For example:

Click on the **Back** button to return to the webpage you last viewed.

`computer output`

When you see text in this style, it indicates text displayed by the computer on the command line. You will see responses to commands you typed in, error messages, and interactive prompts for your input during scripts or programs shown this way. For example:

Use the `ls` command to display the contents of a directory:

```
$ ls
Desktop                about.html       logs          paulwesterberg.png
Mail                   backupfiles      mail          reports
```

The output returned in response to the command (in this case, the contents of the directory) is shown in this style.

`prompt`

A prompt, which is a computer's way of signifying that it is ready for you to input something, will be shown in this style. Examples:

```
$
#
[stephen@maturin stephen]$
leopard login:
```

`user input`

Text that the user has to type, either on the command line, or into a text box on a GUI screen, is displayed in this style. In the following example, **`text`** is displayed in this style:

To boot your system into the text based installation program, you will need to type in the **`text`** command at the `boot:` prompt.

Additionally, we use several different strategies to draw your attention to certain pieces of information. In order of how critical the information is to your system, these items will be marked as note, tip, important, caution, or a warning. For example:

Note

Remember that Linux is case sensitive. In other words, a rose is not a ROSE is not a rOsE.

Tip

The directory `/usr/share/doc` contains additional documentation for packages installed on your system.

Important

If you modify the DHCP configuration file, the changes will not take effect until you restart the DHCP daemon.

Caution

Do not perform routine tasks as root — use a regular user account unless you need to use the root account for system administration tasks.


Warning

If you choose not to partition manually, a server installation will remove all existing partitions on all installed hard drives. Do not choose this installation class unless you are sure you have no data you need to save.

2. More to Come

The *Official Red Hat Linux System Administration Primer* is part of Red Hat's growing commitment to provide useful and timely support to Red Hat Linux users. As new releases of Red Hat Linux are made available, we make every effort to include both new and improved documentation for you.

2.1. Send in Your Feedback

If you spot a typo in the *Official Red Hat Linux System Administration Primer*, or if you have thought of a way to make this manual better, we would love to hear from you. Please submit a report in Bugzilla (http://www.redhat.com/bugzilla) against the component rhl-sap.

Be sure to mention the manual's identifier:

```
rhl-sap(EN)-8.0-Print-RHI (2002-10-01T17:12-0400)
```

If you mention this manual's identifier, we will know exactly which version of the guide you have.

If you have a suggestion for improving the documentation, try to be as specific as possible. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

3. Sign Up for Support

If you have an official edition of Red Hat Linux 8.0, please remember to sign up for the benefits you are entitled to as a Red Hat customer.

You will be entitled to any or all of the following benefits, depending upon the Official Red Hat Linux product you purchased:

- Official Red Hat support — Get help with your installation questions from Red Hat, Inc.'s support team.
- Red Hat Network — Easily update your packages and receive security notices that are customized for your system. Go to http://rhn.redhat.com for more details.
- *Under the Brim: The Official Red Hat E-Newsletter* — Every month, get the latest news and product information directly from Red Hat.

To sign up, go to http://www.redhat.com/apps/activate/. You will find your Product ID on a black, red, and white card in your Official Red Hat Linux box.

To read more about technical support for Official Red Hat Linux, refer to the *Getting Technical Support* Appendix in the *Official Red Hat Linux Installation Guide*.

Good luck, and thank you for choosing Red Hat Linux!

The Red Hat Documentation Team

Background Information

Chapter 1. The Philosophy of System Administration

Although the specifics of being a system administrator may change from platform to platform, there are underlying themes that do not. It is these themes that make up the philosophy of system administration.

Here are those themes:

- Automate everything
- Document everything
- Communicate as much as possible
- Know your resources
- Know your users
- Know your business
- Security cannot be an afterthought
- Plan ahead
- Expect the unexpected

Let us look at each of these themes in more detail.

1.1. Automate Everything

Most system administrators are outnumbered — either by their users, their systems, or both. In many cases, automation is the only way to keep up. In general, anything done more than once should be looked at as a possible candidate for automation.

Here are some commonly automated tasks:

- Free disk space checking and reporting
- Backups
- System performance data collection
- User account maintenance (creation, deletion, etc.)
- Business-specific functions (pushing new data to a Web server, running monthly/quarterly/yearly reports, etc.)

This list is by no means complete; the functions automated by system administrators are only limited by an admin's willingness to write the necessary scripts. In this case, being lazy (and making the computer do more of the mundane work) is actually a good thing.

Automation also gives your users the extra benefit of greater predictability and consistency of service.

Tip

Keep in mind that if you have something that should be automated, it is likely that you are not the first to have that need. Here is where the benefits of open source software really shine — you may be able to leverage someone else's work to automate the very thing that is currently eating up your time. So always make sure you search the Web before writing anything more complex than a small Perl script.

1.2. Document Everything

If given the choice between installing a brand-new server and writing a procedural document on performing system backups, the average system administrator would install the new server every time. While this is not at all unusual, the fact is that you *must* document what you do. Many system administrators will put off doing the necessary documentation for a variety of reasons:

"I will get around to it later."

> Unfortunately, this is usually not true. Even if a system administrator is not kidding themselves, the nature of the job is such that things are usually too chaotic to simply "do it later". Even worse, the longer it is put off, the more that is forgotten, leading to a much less detailed (and therefore, less useful) document.

"Why write it up? I will remember it."

> Unless you are one of those rare individuals with a photographic memory, no, you will not remember it. Or worse, you will remember only half of it, not realizing that you are missing the full story. This leads to wasted time either trying to relearn what you had forgotten, or fixing what you had broken due to not knowing the whole story.

"If I keep it in my head, they will not fire me — I will have job security!"

> While this may work for a while, invariably it leads to less — not more — job security. Think for a moment about what may happen during an emergency. You may not be available; your documentation may save the day by letting someone else fix it. And never forget that emergencies tend to be times when upper management pays close attention. In such cases, it is better to have your documentation be part of the solution than for your unavailability to be part of the problem.
>
> In addition, if you are part of a small but growing organization, eventually there will be a need for another system administrator. How will this person learn to back you up if everything is in your head? Worst yet, not documenting may make you so indispensable that you might not be able to advance your career. You could end up working for the very person that was hired to assist you.

Hopefully you are now sold on the benefits of system documentation. That brings us to the next question: What should you document? Here is a partial list:

Policies

> Policies are written to formalize and clarify the relationship you have with your user community. They make it clear to your users how their requests for resources and/or assistance will be handled. The nature, style, and method of disseminating policies to your user community will vary from organization to organization.

Procedures

> Procedures are any step-by-step sequence of actions that must be taken to accomplish a certain task. Procedures to be documented can include backup procedures, user account management procedures, problem reporting procedures, and so on. Like automation, if a procedure is followed more than once, it is a good idea to document it.

Changes

> A large part of a system administrator's career revolves around making changes — configuring systems for maximum performance, tweaking scripts, modifying printcap files, etc. All of these changes should be documented in some fashion. Otherwise, you could find yourself being completely confused about a change you made several months before.

Some organizations use more complex methods for keeping track of changes, but in most cases a simple revision history at the start of the file being changed is all that is necessary. At a minimum, each entry in the revision history should contain:

- The name or initials of the person making the change
- The date the change was made
- The reason the change was made

This results in concise, yet useful entries:

```
ECB, 12-June-2002 -- Updated entry for new Accounting printer (to
     support the replacement printer's ability to print duplex)
```

1.3. Communicate as Much as Possible

When it comes to your users, you can never communicate too much. Be aware that small system changes you might think are practically unnoticeable could very well completely confuse the administrative assistant in Human Resources.

In general, it is best to follow this somewhat-paraphrased approach used in writing newspaper stories:

1. Tell your users what you are going to do
2. Tell your users what you are doing
3. Tell your users what you have done

Let us look at these three steps in more depth.

1.3.1. Tell Your Users What You Are Going to Do

Make sure you give your users sufficient warning before you do anything. The actual amount of warning will vary according to the type of change (upgrading an operating system demands more lead time than changing the default color of the system login screen), as well as the nature of your user community (more technically adept users may be able to handle changes more readily than users with minimal technical skills).

At a minimum, you should describe:

- The nature of the change
- When it will take place
- Why it is happening
- Approximately how long it should take
- The impact (if any) that the users can expect due to the change
- Contact information should they have any questions or concerns

Here is a hypothetical situation. The Finance department has been experiencing problems with their database server being very slow at times. You are going to bring the server down, upgrade the CPU module to a faster model, and reboot. Once this is done, you will move the database itself to faster, RAID-based storage. Here is one possible announcement for this situation:

System Downtime Scheduled for Friday Night

> Starting this Friday at 6pm (midnight for our associates in Berlin), all financial applications will be unavailable for a period of approximately four hours.
>
> During this time, changes to both the hardware and software on the Finance database server will be performed. These changes should greatly reduce the time required to run the Accounts Payable and Accounts Receivable applications, and the weekly Balance Sheet report.
>
> Other than the change in runtime, most people will notice no other change. However, those of you that have written your own SQL queries should be aware that the layout of some indices will change. This is documented on the company intranet website, on the Finance page.
>
> Should you have any questions, comments, or concerns, please contact System Administration at extension 4321.

A few points are worth noting here:

- Effectively communicate the start and duration of any downtime that might be involved in the change.
- Make sure you give the time of the change in such a way that it is useful to *all* users, no matter where they may be located.
- Use terms that your users will understand. The people impacted by this work do not care that the new CPU module has twice as much cache, or that the database will live on a RAID 5 logical volume.

1.3.2. Tell Your Users What You Are Doing

This step is primarily a last-minute warning of the impending change; as such, it should be a brief repeat of the first message, though with the impending nature of the change made more apparent ("The system upgrade will take place TOMORROW."). This is also a good place to publicly answer any questions you may have received as a result of the first message.

Continuing our hypothetical example, here is one possible last-minute warning:

System Downtime Scheduled for Tonight

> Reminder: The system downtime announced this past Monday will take place as scheduled tonight at 6pm (midnight for the Berlin office). You can find the original announcement on the company intranet website, on the System Administration page.
>
> Several people have asked whether they should stop working early tonight to make sure their work is backed up prior to the downtime. This will not be necessary, as the work being done tonight will not impact any work done on your personal workstations.

Your users have been alerted; now you are ready to actually do the work.

1.3.3. Tell Your Users What You Have Done

After you have finished making the changes, you *must* tell them what you have done. Again, this should be a summary of the previous messages (invariably someone will not have read them).

However, there is one important addition that you must make. It is vital that you give your users the current status. Did the upgrade not go as smoothly as planned? Was the new storage server only able to serve the systems in Engineering, and not in Finance? These types of issues must be addressed here.

Of course, if the current status differs from what you communicated previously, you should make this point clear, and describe what will be done (if anything) to arrive at the final solution.

In our hypothetical situation, the downtime had some problems. The new CPU module did not work; a call to the system's manufacturer revealed that a special version of the module is required for in-the-field upgrades. On the plus side, the migration of the database to the RAID volume went well (even though it took a bit longer than planned due to the problems with the CPU module.

Here is one possible announcement:

System Downtime Complete

> The system downtime scheduled for Friday night (please see the System Administration page on the company intranet website) has been completed. Unfortunately, hardware issues prevented one of the tasks from being completed. Due to this, the remaining tasks took longer than the originally-schedule four hours. Instead, all systems were back in production by midnight (6am Saturday for the Berlin office).
>
> Because of the remaining hardware issues, performance of the AP, AR, and the Balance Sheet report will be slightly improved, but not to the extent originally planned. A second downtime will be announced and scheduled as soon as the nature of the hardware issues is clear.
>
> Please note that the downtime did change some database indices; people that have written their own SQL queries should consult the Finance page on the company intranet website. Please contact System Administration at extension 4321 with any questions.

With this kind of information, your users will have sufficient background to continue their work, and to understand how the changes will impact them.

1.4. Know Your Resources

System administration is mostly a matter of balancing available resources against the people and programs that use those resources. Therefore, your career as a system administrator will be a short and stress-filled one unless you fully understand the resources you have at your disposal.

Some of the resources are ones that seem pretty obvious:

- System resources, such as available processing power, memory, and disk space
- Network bandwidth
- Available money from the IT budget

But some may not be so obvious:

- The services of operations personnel, other admins, or even an administrative assistant
- Time (often of critical importance when the time involves things such as the amount of time during which system backups may take place)
- Knowledge — whether it is stored in books, system documentation, or the brain of a person that has worked at the company for the past twenty years

The important thing to note is that it is highly valuable to take a complete inventory of those resources that are available to you, and to *keep it current* — a lack of "situational awareness" when it comes to available resources can often be worse than *no* awareness.

1.5. Know Your Users

Although some people bristle at the term "users" (perhaps due to some system administrators' use of the term in a derogatory manner), it is used here with no such meaning implied. Users are simply those people that use the systems and resources for which you are responsible. As such, they are central to your ability to successfully administer your systems; without understanding your users, how can you understand the system resources they will require?

For example, consider a bank teller. A bank teller will use a strictly-defined set of applications, and requires little in the way of system resources. A software engineer, on the other hand, may use many different applications, and will always welcome more system resources (for faster build times). Two entirely different users with two entirely different needs.

Make sure you learn as much about your users as you can.

1.6. Know Your Business

Whether you work for a large, multinational corporation or a small community college, you must still understand the nature of the business environment in which you work. This can be boiled down to one question:

What is the purpose of the systems you administer?

The key point here is to understand your systems' purpose in a more global sense:

- Applications that must be run within certain time frames, such as at the end of a month, quarter, or year
- The times during which system maintenance may be done
- New technologies that could be used to resolve long-standing business problems

By taking into account your organization's business, you will find that your day-to-day decisions will be better for your users. And for you.

1.7. Security Cannot be an Afterthought

No matter what you might think about the environment in which your systems are running, you cannot take security for granted. Even standalone systems not connected to the Internet may be at risk (although obviously the risks will be different from a system that is more connected to the outside world).

Therefore, it is extremely important to consider the security implications of everything that you do. The following lists illustrates the different kinds of issues that you should consider:

- The nature of possible threats to each of the systems under your care
- The location, type, and value of data on those systems
- The type and frequency of authorized access to the systems (and their data)

While you are thinking about security, do not make the mistake of assuming that possible intruders will only attack your systems from outside of your company. Many times the perpetrator is someone within the company. So the next time you walk around the office, look at the people around you and ask yourself this question:

What would happen if *that* person were to attempt to subvert our security?


Note

This does *not* mean that you should treat your coworkers as if they are criminals. It just means that you should look at the type of work that each person performs, and determine what types of security breaches a person in that position could perpetrate, if they were so inclined.

1.8. Plan Ahead

A system administrator that took all the previous advice to heart and did their best to follow it would be a fantastic system administrator — for a day. Eventually, the environment will change, and one day our fantastic administrator would be caught flat-footed. The reason? Our fantastic administrator failed to plan ahead.

Certainly no one can predict the future with 100% accuracy. However, with a bit of awareness it is easy to read the signs of many changes:

- An offhand mention of a new project gearing up during that boring weekly staff meeting is a sure sign that you will likely need to support new users
- Talk of an impending acquisition means that you may end up being responsible for new (and possibly incompatible) systems in one or more remote locations

Being able to read these signs (and to effectively respond to them) will make life easier for you and your users.

1.9. Expect the Unexpected

While the phrase "expect the unexpected" is trite, it reflects an underlying truth that all system administrators must understand:

There *will* be times when you are caught off-guard.

After becoming comfortable with this uncomfortable fact of life, what can a concerned system administrator do? The answer lies in flexibility; by performing your job in such a way as to give you (and your users) the most options possible. Take, for example, the issue of disk space. Given that never having sufficient disk space seems to be as much a physical law as the law of gravity, it is reasonable to assume that at some point you will be confronted with a desperate need for additional disk space *right now*.

What would a system administrator who expects the unexpected do in this case? Perhaps it is possible to keep a few disk drives sitting on the shelf as spares in case of hardware problems[1]. A spare of this type could be quickly deployed[2] on a temporary basis to address the short-term need for disk space, giving time to more permanently resolve the issue (by following the standard procedure for procuring additional disk drives, for example).

By trying to anticipate problems before they occur, you will be in a position to respond more quickly and effectively than if you let yourself be surprised.

1. And of course a system administrator that expects the unexpected would naturally use RAID (or related technologies) to lessen the impact of a disk drive that fails during production.
2. Again, system administrators that think ahead will configure their systems to make it as easy as possible to quickly add a new disk drive to the system.

1.10. In Conclusion...

While everything discussed in this chapter may seem like a lot of additional work that takes away from the "real" work of administering systems, actually the opposite is true; only by keeping this philosophy in mind will you give your users the service they deserve, and reach your full potential as a system administrator.

Resource Management

Chapter 2.

Resource Monitoring

As stated earlier, a great deal of system administration revolves around resources and their efficient use. By balancing various resources against the people and programs that use those resources, you will waste less money and make your users as happy as possible. However, this leaves two questions:

What are resources?

And:

How do I know what resources are being used (and to what extent)?

The purpose of this chapter is to enable you to answer these questions by helping you to learn more about resources and how their utilization can be monitored.

2.1. Basic Concepts

Before you can monitor resources, you first have to know what resources there are to monitor. All systems have the following resources available:

- CPU power (and the bandwidth it enables)
- Memory
- Storage

We will study these resources in more depth in the following chapters. However, for the time being all you need to keep in mind is that these resources have a direct impact on system performance, and therefore, on your users' productivity and happiness.

At its simplest, resource monitoring is nothing more than obtaining information concerning the utilization of one or more system resources.

However, it is rarely this simple. First, one must take into account the resources to be monitored. Then it is necessary to look at each system to be monitored, paying particular attention to each system's situation.

The systems you will be monitoring will fall into one of two categories:

- The system is currently experiencing performance problems at least part of time and you would like to improve its performance
- The system is currently running well and you would like it to stay that way.

The first category means that you should monitor resources from a system performance perspective, while the second category means that you should monitor system resources from a capacity planning perspective.

Because each perspective has its own unique requirements, we will now look at each category in more depth.

2.2. System Performance Monitoring

As stated above, system performance monitoring is normally done in response to a performance problem. Either the system is running too slowly, or programs (and sometimes even the entire system) fail to run at all. In either case, performance monitoring is normally done as the first and last two steps of a three-step process:

- Monitoring to identify the nature and scope of the resource shortages that are causing the performance problems
- The data produced from monitoring is analyzed and a course of action (normally performance tuning and/or the procurement of additional hardware) is taken to resolve the problem
- Monitoring to ensure that the performance problem has been resolved

Because of this, performance monitoring tends to be relatively short-lived in duration, and more detailed in scope.


Note

System performance monitoring is often an iterative process, with these steps being repeated several times to arrive at the best possible system performance. The primary reason for this is that system resources and their utilization tend to be highly interrelated, meaning that often the elimination of one resource bottleneck simply uncovers another one.

2.3. Monitoring System Capacity

Monitoring system capacity is done as part of an ongoing capacity planning program. Capacity planning uses long-term resource monitoring to determine rates of change in the utilization of system resources. Once these rates of change are known, it becomes possible to conduct more accurate long-term planning regarding the procurement of additional resources.

Monitoring done for capacity planning purposes is different from performance monitoring in two ways:

- The monitoring is done on a more-or-less continuous basis
- The monitoring is usually not as detailed

The reason for these differences stems from the goals of a capacity planning program. Capacity planning requires a "big picture" view; short-term or anomalous resource usage is of little concern. Instead, data collected over a period of time makes it possible to start to categorize resource utilization in terms of the impact of changes in workload on resource availability. In more narrowly-defined environments (where only one application is run, for example) it is possible to model the application's impact on system resources, leading to the ability to determine, for example, the impact of five more customer service representatives running the customer service application during the busiest time of the day.

Next, we will look at tools that make it possible to observe system resource utilization.

2.4. Resource Monitoring Tools

Red Hat Linux comes with a variety of resource monitoring tools. While there are more than those listed here, these tools are representative in terms of functionality. The tools we will look at are:

- `free`
- `top`
- The **sysstat** suite of resource monitoring tools

Let us look at each one in more detail.

2.4.1. free

The free command displays memory utilization data. Here is an example of its output:

```
             total       used       free     shared    buffers     cached
Mem:        255508     240268      15240          0       7592      86188
-/+ buffers/cache:     146488     109020
Swap:       530136      26268     503868
```

The Mem: row displays physical memory utilization, while the Swap: row displays the utilization of the system swap space, while the -/+ buffers/cache: row displays the amount of physical memory currently devoted to system buffers.

Since free by default only displays memory utilization information once, it is only useful for very short-term monitoring. Although free has the ability to repetitively display memory utilization figures via its -s option, the output simply scrolls, making it difficult to easily see changes in memory utilization.

Tip

A better solution would be to run free using the watch command. For example, to display memory utilization every two seconds, use this command:

```
watch free
```

You can control the delay between updates by using the -n option, and can cause any changes between updates to be highlighted by using the -d option, as in the following command:

```
watch -n 1 -d free
```

For more information, see the watch man page.

The watch command will run until interrupted with [Ctrl]-[C]. Make sure you remember watch; it can come in handy in many situations.

2.4.2. top

While free displays only memory-related information, the top command does a little bit of everything. CPU utilization, process statistics, memory utilization — top does it all. In addition, unlike the free command, top's default behavior is to run continuously; no need for the watch command here. Here is a sample display:

```
11:13am  up 1 day, 31 min,  5 users,  load average: 0.00, 0.05, 0.07
89 processes: 85 sleeping, 3 running, 1 zombie, 0 stopped
CPU states:  0.5% user,  0.7% system,  0.0% nice, 98.6% idle
Mem:  255508K av, 241204K used,  14304K free,    0K shrd,   16604K buff
Swap: 530136K av,  56964K used, 473172K free                64724K cached

  PID USER     PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM   TIME COMMAND
 8532 ed       16   0  1156 1156   912 R     0.5  0.4   0:11 top
 1520 ed       15   0  4084 3524  2752 S     0.3  1.3   0:00 gnome-terminal
 1481 ed       15   0  3716 3280  2736 R     0.1  1.2   0:01 gnome-terminal
 1560 ed       15   0 11216  10M  4256 S     0.1  4.2   0:18 emacs
    1 root     15   0   472  432   416 S     0.0  0.1   0:04 init
    2 root     15   0     0    0     0 SW    0.0  0.0   0:00 keventd
    3 root     15   0     0    0     0 SW    0.0  0.0   0:00 kapmd
    4 root     34  19     0    0     0 SWN   0.0  0.0   0:00 ksoftirqd_CPU0
    5 root     15   0     0    0     0 SW    0.0  0.0   0:00 kswapd
```

```
    6 root     25   0     0    0     0 SW    0.0  0.0   0:00 bdflush
    7 root     15   0     0    0     0 SW    0.0  0.0   0:00 kupdated
    8 root     25   0     0    0     0 SW    0.0  0.0   0:00 mdrecoveryd
   12 root     15   0     0    0     0 SW    0.0  0.0   0:00 kjournald
   91 root     16   0     0    0     0 SW    0.0  0.0   0:00 khubd
  185 root     15   0     0    0     0 SW    0.0  0.0   0:00 kjournald
  186 root     15   0     0    0     0 SW    0.0  0.0   0:00 kjournald
  576 root     15   0   712  632   612 S     0.0  0.2   0:00 dhcpcd
```

The display is separated into two main parts. The top section contains information related to overall system status, process counts, along with memory and swap utilization. The lower section displays process-level statistics, the exact nature of which can be controlled while `top` is running.

For more information on `top`, refer to the `top` man page.

Warning

Although `top` looks like a simple display-only program, this is not the case. If you are logged in as root, it is possible to change the priority and even kill any process on your system.

Therefore, make sure you read the `top` man page before using it.

2.4.2.1. `gnome-system-monitor` — A Graphical `top`

If you are more comfortable with graphical user interfaces, `gnome-system-monitor` may be more to your liking. Like `top`, `gnome-system-monitor` displays information related to overall system status, process counts, memory and swap utilization, and process-level statistics.

However, `gnome-system-monitor` goes a step further by including displays disk space utilization — something that `top` does not do at all.

2.4.3. The sysstat Suite of Resource Monitoring Tools

While the previous tools may be helpful for gaining more insight into system performance over very short time frames, they are of little use beyond providing a snapshot of system resource utilization. In addition, there are aspects of system performance that cannot be easily monitored using such simplistic tools.

Therefore, a more sophisticated tool is necessary. **sysstat** is such a tool.

sysstat contains the following commands related to collecting I/O and CPU statistics:

`iostat`

Displays I/O statistics for one or more disk drives. The statistics returned can include read and write rates per second, average wait, service, and CPU utilization, and more.

`mpstat`

Displays CPU statistics.

However, the most versatile and sophisticated tools that are part of **sysstat** are those related to the `sar` command. Collectively these tools:

- Collect system resource utilization data
- Create daily reports of system resource utilization

- Allow the graphical viewing of system resource utilization data

The tools that perform these tasks are:

`sadc`

`sadc` is known as the system activity data collector. It collects system resource utilization information and writes it to files in the `/var/log/sa/` directory. The files are named `sa`<*dd*>, where <*dd*> is the current day's two-digit date.

`sa1`

`sa1` is a script that runs `sadc` to perform the actual data collection, and is run by `cron` at regular intervals throughout the day.

`sar`

`sar` produces reports from the files created by `sadc`. The report files written to `/var/log/sa/`, and are named `sar`<*dd*>, where <*dd*> is the two-digit representations of the previous day's date.

`sa2`

`sa2` is a script that uses `sar` to write a daily system resource utilization report. `sa2` is run by `cron` once at the end of each day.

`isag`

`isag` graphically displays data collected by `sadc`.

`sa`

Summarizes system accounting information.

The **sysstat** tools should be part of every system administrator's resource monitoring tool bag.

redhat.

Chapter 3.

Bandwidth and Processing Power

Of the two resources discussed in this chapter, one (bandwidth) is often hard for the new system administrator to understand, while the other (processing power) is usually a much easier concept to grasp.

Additionally, it may seem that these two resources are not that closely related — why group them together?

The reason for addressing both resources together is that these resources are based on the hardware that tie directly into a computer's ability to move and process data. As such, their relationship is often interrelated.

3.1. Bandwidth

At its simplest, bandwidth is simply the capacity for data transfer — in other words how much data can be moved from one point to another in a given amount of time. Having point-to-point data communication implies two things:

- A set of electrical conductors used to make low-level communication possible
- A protocol to facilitate the efficient and reliable communication of data

There are two types of system components that meet these requirements:

- Buses
- Datapaths

In the following sections, we will explore both in more detail.

3.1.1. Buses

As stated above, buses enable point-to-point communication, and use some sort of protocol to ensure that all communication takes place in a controlled manner. However, buses have other distinguishing features:

- Standardized electrical characteristics (such as the number of conductors, voltage levels, signaling speeds, etc.)
- Standardized mechanical characteristics (such as the type of connector, card size, etc.)
- Standardized protocol

The word "standardized" is important because buses are the primary way in which different system components are connected together.

In many cases, buses allow the interconnection of hardware that is made by multiple manufacturers; without standardization, this would not be possible. However, even in situations where a bus is proprietary to one manufacturer, standardization is important because it allows that manufacturer to more easily implement different components by using a common interface — the bus.

3.1.1.1. Examples of Buses

No matter where in a computer system you look, you will see buses. Here are a few of the more common ones:

- Mass storage buses (IDE and SCSI)
- Networks (instead of an intra-system bus, networks can be thought of as an *inter*-system bus)
- Memory buses
- Expansion buses (PCI, ISA, USB)

3.1.2. Datapaths

Datapaths can be harder to identify but, like buses, they are everywhere. Also like buses, datapaths enable point-to-point communication. However, unlike buses, datapaths:

- Use a simpler protocol (if any)
- Have little (if any) mechanical standardization

The reason for these differences is that datapaths are normally internal to some system component, and are not used to facilitate the ad-hoc interconnection of different components. As such, datapaths are highly optimized for a particular situation, where speed and low cost are preferred over general-purpose flexibility.

3.1.2.1. Examples of Datapaths

Here are some typical datapaths:

- CPU to on-chip cache datapath
- Graphics processor to video memory datapath

3.1.3. Potential Bandwidth-Related Problems

There are two ways in which bandwidth-related problems may occur (for either buses or datapaths):

1. The bus or datapath may represent a shared resource. In this situation, high levels of contention for the bus will reduce the effective bandwidth available for all devices on the bus.

   A SCSI bus with several highly-active disk drives would be a good example of this. The highly-active disk drives will saturate the SCSI bus, leaving little bandwidth available for any other device on the same bus. The end result is that all I/O to any of the devices on this bus will be slow, even if the device itself is not overly active.

2. The bus or datapath may be a dedicated resource with a fixed number of devices attached to it. In this case, the electrical characteristics of the bus (and to some extent the nature of the protocol being used) limit the available bandwidth. This is usually more the case with datapaths than with buses.

   This is one reason why graphics adapters tend to perform more slowly when operating at higher resolutions and/or color depths; there is more data that must be passed between the datapath connecting video memory and the graphics processor.

3.1.4. Potential Bandwidth-related Solutions

Fortunately, bandwidth-related problems can be addressed. In fact, there are several approaches you can take to address bandwidth-related problems:

- Increase the capacity
- Spread the load
- Reduce the load

In the following sections, we will explore each approach in more detail.

3.1.4.1. Increase the Capacity

The obvious solution to insufficient bandwidth is to increase it somehow. However, this is usually an expensive proposition. Consider, for example, a SCSI controller and its overloaded bus. In order to increase its bandwidth, the SCSI controller, and likely all devices attached to it, would need to be replaced. If the SCSI controller is a separate card, this would be a relatively straightforward process, but if the SCSI controller is part of the system's motherboard, it becomes much more difficult to justify the economics of such a change.

3.1.4.2. Spread the Load

Another approach is to more evenly distribute the bus activity. In other words, if one bus is overloaded, and another is idle, perhaps the situation would be improved by moving some of the load to the idle bus.

As a system administrator, this is the first approach you should consider, as often there are additional buses already present in your system. For example, most PCs include at least two IDE *channels* (which is just another name for a bus). If you have two IDE disk drives and two IDE channels, why should the drives both be on the same channel?

Even if your system configuration does not include additional buses, spreading the load might still be a reasonable approach. The hardware expenditures to do so would be less expensive than replacing an existing bus with higher-capacity hardware.

3.1.4.3. Reduce the Load

At first glance, reducing the load and spreading the load appear to be different sides of the same coin. After all, when one spreads the load, it acts to reduce the load (at least on the overloaded bus), correct?

While this viewpoint is correct, it is not the same as reducing the load *globally*. The key here is to determine if there is some aspect of the system load that is causing this particular bus to be overloaded. For example, is a network heavily loaded due to activities that are unnecessary? Perhaps a small temporary file is the recipient of heavy read/write I/O. If that temporary file was created on a networked file server, a great deal of network traffic could be eliminated by simply working with the file locally.

3.1.5. In Summary...

All system administrators should be aware of bandwidth, and how system configuration and usage impacts available bandwidth. Unfortunately, it is not always apparent what is a bandwidth-related problem and what is not. Sometimes, the problem is not the bus itself, but one of the components attached to the bus.

For example, consider a SCSI adapter that is connected to a PCI bus, and providing a SCSI bus. However, if there are performance problems with SCSI I/O, it might be the result of a poorly-performing SCSI adapter, even though the SCSI and PCI buses are nowhere near their bandwidth capabilities.

3.2. Processing Power

Often known as CPU power, CPU cycles, and various other names, processing power is the ability of a computer to manipulate data. Processing power varies with the architecture (and clock speed) of the CPU — usually CPUs with higher clock speeds and those supporting larger word sizes have more processing power than slower CPUs supporting smaller word sizes.

3.2.1. Facts About Processing Power

There are two main facts about processing power that you should keep in mind:

- It is fixed
- It cannot be stored

Processing power is fixed, in that the CPU can only go so fast. For example, if you need to add two numbers together (an operation that takes only one machine instruction on most architectures), a particular CPU can do it at one speed, and one speed only. With few exceptions, it is not even possible to *slow* the rate at which a CPU processes instructions.

Processing power is also fixed in another way: it is finite. That is, there are limits to the CPU performance you can put into any given computer. Some systems are capable of supporting a wide range of CPU speeds, while others may not be upgradeable at all[1].

Processing power cannot be stored for later use. In other words, if a CPU can process 100 million instructions in one second, one second of idle time equals 100 million instructions that have been wasted.

If we take these facts and look at them from a slightly different perspective, a CPU "produces" a stream of executed instructions at a fixed rate. And if the CPU produces executed instructions, that means that something else must "consume" them.

3.2.2. Consumers of Processing Power

There are two main consumers of processing power:

- Applications
- The operating system itself

3.2.2.1. Applications

The most obvious consumers of processing power are the applications and programs you want the computer to run for you. From a spreadsheet to a database, these are the reasons you have a computer.

A single-CPU system can only run one thing at any given time. Therefore, if your application is running, everything else on the system is not. And the opposite is, of course, true — if something other than your application is running, then your application is doing nothing.

1. This situation leads to what is humorously termed as a *forklift upgrade*, which means a complete replacement of the computer.

But how is it that many different applications can run at once under Red Hat Linux? The answer is that Red Hat Linux is a multitasking operating system. In other words, it creates the illusion that many things are going on simultaneously when in fact that is impossible. The trick is to give each process a fraction of a second's worth of time running on the CPU before giving the CPU to another process for another fraction of a second. If these *context switches* happen quickly enough, the illusion of multiple applications running simultaneously is achieved.

Of course, applications do other things than manipulate data using the CPU. They may wait for user input as well as performing I/O to many devices, including disk drives and graphics displays. When these events take place, the application does not need the CPU. At these times, the CPU can be used for other processes running other applications.

In addition, the CPU can be used by another consumer of processing power: the operating system itself.

3.2.2.2. The Operating System

It is difficult to determine how much processing power is consumed by the operating system. The reason for this is that operating systems use a mixture of process-level and system-level code to perform their work. While it is easy to use `top` to see what the process running `syslogd` is doing (for example), it is not so easy to see how much processing power is being consumed by I/O-related processing.

In general, it is possible to divide this kind of operating system overhead into two types:

- Operating system housekeeping
- Process-related activities

Operating system housekeeping includes activities such as process scheduling and memory management, while process-related activities include any processes that support the operating system itself (including system daemons such as `syslogd`, `klogd`, etc.).

3.2.3. Improving a CPU Shortage

When there is insufficient processing power available for the work that needs to be done, you have two options:

- Reducing the load
- Increasing Capacity

3.2.3.1. Reducing the Load

Reducing the CPU load is something that can be done with no expenditure of money. The trick is to identify those aspects of the system load that are under your control and can be cut back. There are three areas to focus on:

- Reducing operating system overhead
- Reducing application overhead
- Eliminating applications entirely

3.2.3.1.1. Reducing Operating System Overhead

In order to reduce operating system overhead, you will have to look at your current system load, and determine what aspects of it result in inordinate amounts of overhead. These areas could include:

- Reducing the need for frequent process scheduling
- Lowering the amount of I/O performed

Do not expect miracles; in a reasonably-well configured system, it is unlikely that you will see much of a performance increase by trying to reduce operating system overhead. This is due to the fact that a reasonably-well configured system will, by definition, result in a minimal amount of overhead. However, if your system is running with too little RAM for instance, you may be able to reduce overhead by alleviating the RAM shortage.

3.2.3.1.2. Reducing Application Overhead

Reducing application overhead simply means making sure that the application has everything it needs to run well. Some applications exhibit wildly different behaviors under different environments — an application may become highly compute-bound while processing certain types of data, but not others.

The point to keep in mind here is that you must understand the applications running on your system if you are to enable them to run as efficiently as possible. Often this entails working with your users, and/or your organization's developers, to help uncover ways in which the applications can be made to run more efficiently.

3.2.3.1.3. Eliminating Applications Entirely

Depending on your organization, this approach might not be available to you, as it often is not a system administrator's job to dictate which application will and will not be run. However, if you can identify any applications that are known "CPU hogs", you might be able to retire them.

Doing this will likely involve more than just yourself. The affected users should certainly be a part of this process; in many cases they may have the knowledge and the political power to make the necessary changes to the application lineup.

Tip

Keep in mind that an application may not need to be eliminated from every system in your organization. You might be able to move a particularly CPU-hungry application from an overloaded system to another system that is nearly idle.

3.2.3.2. Increasing Capacity

Of course, if it is not possible to reduce the demand for processing power, you will have to find ways of increasing the processing power that is available. To do so will cost money, but it can be done.

3.2.3.2.1. Upgrading the CPU

The most straightforward approach is to determine if your system's CPU can be upgraded. The first step is to see if the current CPU can be removed. Some systems (primarily laptops) have CPUs that are soldered in place, making an upgrade impossible. The rest, however, have socketed CPUs, making upgrades theoretically possible.

Next, you will have to do some research to determine if a faster CPU exists for your system configuration. For example, if you currently have a 1GHz CPU, and a 2GHz unit of the same type exists, an upgrade might be possible.

Finally, you must determine the maximum clock speed supported by your system. To continue the example above, even if a 2GHz CPU of the proper type exists, a simple CPU swap is not an option if your system only supports processors running at 1GHz or below.

Should you find that you cannot install a faster CPU in your system, your options may be limited to changing motherboards, or even the more expensive forklift upgrade mentioned earlier.

However, some system configurations make a slightly different approach possible. Instead of replacing the current CPU, why not just add another one?

3.2.3.2.2. Is Symmetric Multiprocessing Right for You?

Symmetric multiprocessing (also known as SMP) makes it possible for a computer system to have more than one CPU sharing all system resources. This means that, unlike a uniprocessor system, an SMP system may actually have more than one process running at the same time.

At first glance, this seems like any system administrator's dream. First and foremost, SMP makes it possible to increase a system's CPU power even if CPUs with faster clock speeds are not available — just by adding another CPU. However, this flexibility comes with some caveats.

The first caveat is that not all systems are capable of SMP operation. Your system must have a motherboard designed to support multiple processors.

The second caveat is that SMP increases system overhead. This makes sense if you stop to think about it; with more CPUs to schedule work for, the operating system will require more CPU cycles for overhead. Another aspect to this is that with multiple CPUs, there can be more contention for system resources. Because of these factors, upgrading a dual-processor system to a quad-processor unit will not result in a 100% increase in available CPU power. In fact, depending on the actual hardware, the workload, and the processor architecture, it is possible to reach a point where the addition of another processor could actually *reduce* system performance.

Another point to keep in mind is that SMP will not help workloads that consist of one monolithic application with a single stream of execution. In other words, if a large compute-bound simulation program runs as one process and with no threads, it will not run any faster on an SMP system than on a single-processor machine. In fact, it may even run somewhat slower, due to the increased overhead SMP brings. For these reasons, many system administrators feel that when it comes to CPU power, single stream processing power is the way to go.

While this discussion seems to indicate that SMP is never a good idea, there are circumstances in which it makes sense. For example, environments running multiple highly compute-bound applications are good candidates for SMP. The reason for this is that applications that do nothing but compute for long periods of time keep contention between active processes (and therefore, the operating system overhead) to a minimum, while the processes themselves will keep every CPU busy.

One other thing to keep in mind about SMP is that the performance of an SMP system tends to degrade more gracefully as the system load increases. This does make SMP systems popular in server and multi-user environments, as the ever-changing process mix will impact the system-wide load less on a multi-processor machine.

Chapter 4.

Physical and Virtual Memory

All present-day general-purpose computers are of the type known as *stored program computers*. As the name implies, stored program computers can load instructions (groups of which make up programs) into some type of internal storage and can subsequently execute those instructions.

Stored program computers also use the same storage for data. This is in contrast to computers that use their hardware configuration to control their operation (such as older plugboard-based computers).

The place where programs were stored on the first stored program computers went by a variety of names and used a variety of different technologies, from spots on a cathode ray tube, to pressure pulses in columns of mercury. Fortunately, present-day computers use technologies with greater storage capacity and much smaller size than ever before.

4.1. Storage Access Patterns

One thing to keep in mind throughout this chapter is that computers tend to access storage in certain ways. In fact, most storage access tends to exhibit one (or both) of the following attributes:

- Access tends to be sequential
- Access tends to be localized

Let us look at these points in a bit more detail.

Sequential access means that, if address N is accessed by the CPU, it is highly likely that address N+1 will be accessed next. This makes sense, as most programs consist of large sections of instructions that execute one after the other.

Localized access means that, if address X is accessed, it is likely that other addresses surrounding X will also be accessed in the future.

These attributes are crucial, because it allows smaller, faster storage to effectively buffer slower, larger storage. This is the basis for implementing virtual memory. But before we can discuss virtual memory, we must look at the various storage technologies currently in use.

4.2. The Storage Spectrum

Present-day computers actually use a variety of storage technologies. Each technology is geared toward a specific function, with speeds and capacities to match. These technologies are:

- CPU registers
- Cache memory
- RAM
- Hard drives
- Off-line backup storage (tape, optical disk, etc.)

In terms of each technologies capabilities and cost, these technologies form a spectrum. For example, CPU registers are:

- Very fast (access times of a few nanoseconds)
- Low capacity (usually less than 200 bytes)

- Very limited expansion capabilities (A change in CPU architecture would be required)
- Expensive (more than one dollar/byte)

However, at the other end of the spectrum, off-line backup storage is:

- Very slow (access times may be measured in days, if the backup media must be shipped long distances)
- Very high capacity (10s - 100s of gigabytes)
- Essentially unlimited expansion capabilities (limited only by the floorspace needed to house the backup media)
- Very inexpensive (fractional cents/byte)

By using different technologies with different capabilities, it is possible to fine-tune system design for maximum performance at the lowest possible cost.

4.2.1. CPU Registers

Every present-day CPU design includes registers for a variety of purposes, from storing the address of the currently-executed instruction to more general-purpose data storage and manipulation. CPU registers run at the same speed as the rest of the CPU; otherwise, they would be a serious bottleneck to overall system performance. The reason for this is that nearly all operations performed by the CPU involve the registers in one way or another.

The number of CPU registers (and their uses) are strictly dependent on the architectural design of the CPU itself. There is no way to change the number of CPU registers, short of migrating to a CPU with a different architecture. For these reasons, the number of CPU registers can be considered a constant (unchangeable without great pain).

4.2.2. Cache Memory

The purpose of cache memory is to act as a buffer between the very limited, very high-speed CPU registers and the relatively slower and much larger main system memory — usually referred to as RAM[1]. Cache memory has an operating speed similar to the CPU itself, so that when the CPU accesses data in cache the CPU is not kept waiting for the data.

Cache memory is configured such that, whenever data is to be read from RAM, the system hardware first checks to see if the desired data is in cache. If the data is in cache, it is quickly retrieved, and used by the CPU. However, if the data is not in cache, the data is read from RAM and, while being transferred to the CPU, is also placed in cache (in case it will be needed again). From the perspective of the CPU, all this is done transparently, so that the only difference between accessing data in cache and accessing data in RAM is the amount of time it takes for the data to be returned.

In terms of storage capacity, cache is much smaller than RAM. Therefore, not every byte in RAM can have its own location in cache. As such, it is necessary to split cache up into sections that can be used to cache different areas of RAM and to have a mechanism that allows each area of cache to cache different areas of RAM at different times. However, given the sequential and localized nature of storage access, a small amount of cache can effectively speed access to a large amount of RAM.

When writing data from the CPU, things get a bit more complicated. There are two different approaches that can be used. In both cases, the data is first written to cache. However, since the purpose of cache is to function as a very fast copy of the contents of selected portions of RAM, any time

1. While "RAM" is an acronym standing for "Random Access Memory," and that term could easily apply to any storage technology that allowed the non-sequential access of stored data, when system administrators talk about RAM they invariably mean main system memory.

a piece of data changes its value, that new value must be written to both cache memory and RAM. Otherwise, the data in cache and the data in RAM will no longer match.

The two approaches differ in how this is done. One approach, known as *write-through* cache, immediately writes the modified data to RAM. *Write-back* cache, however, delays the writing of modified data back to RAM; in this way, should the data be modified again it will not be necessary to undergo several slow data transfers to RAM.

Write-through cache is a bit simpler to implement; for this reason it is often seen. Write-back cache is a bit trickier to implement, as in addition to storing the actual data, it is necessary to maintain some sort of flag that denotes that the cached data is clean (the data in RAM is the same as the data in cache), or dirty (the data in RAM is not the same as the data in cache). It is also necessary to implement a way of periodically flushing dirty cache entries back to RAM.

4.2.2.1. Cache Levels

Cache subsystems in present-day computer designs may be multi-level; that is, there might be more than one set of cache between the CPU and main memory. The cache levels are often numbered, with lower numbers being closer to the CPU. Many systems have two cache levels:

- L1 cache is often directly on the CPU chip itself and runs at the same speed as the CPU
- L2 cache is often part of the CPU module, runs at CPU speeds (or nearly so), and is usually a bit larger and slower than L1 cache

Some systems (normally high-performance servers) also have L3 cache, which is usually part of the system motherboard. As might be expected, L3 cache would be larger (and most likely slower) than L2 cache. In either case, the goal of all cache subsystems — whether single- or multi-level — is to reduce the average access time to the RAM.

4.2.3. Main Memory — RAM

RAM makes up the bulk of electronic storage on present-day computers. It is used as storage for both data and programs while those data and programs are in use. The speed of RAM in most systems today lies between the speeds of cache memory and that of hard drives and is much closer to the former than the latter.

The basic operation of RAM is actually quite straightforward. At the lowest level, there are the RAM chips — integrated circuits that do the actual "remembering." These chips have four types of connections to the outside world:

- Power connections (to operate the circuitry within the chip)
- Data connections (to enable the transfer of data into or out of the chip)
- Read/Write connections (to control whether data is to be stored into or retrieved from the chip)
- Address connections (to determine where in the chip the data should be read/written)

Here are the steps required to store data in RAM:

- The data to be stored is presented to the data connections.
- The address at which the data is to be stored is presented to the address connections.
- The read/write connection to set to write mode.

Retrieving data is just as simple:

- The address of the desired data is presented to the address connections.
- The read/write connection is set to read mode.
- The desired data is read from the data connections.

While these steps are simple, they take place at very high speeds, with the time spent at each step measured in nanoseconds.

Nearly all RAM chips created today are sold as *modules*. Each module consists of a number of individual RAM chips attached to a small circuit board. The mechanical and electrical layout of the module adhere to various industry standards, making it possible to purchase memory from a variety of vendors.


Note

The main benefit to a system that uses industry-standard RAM modules is that it tends to keep the cost of RAM low, due to the ability to purchase the modules from more than just the system manufacturer.

Although most computers use industry-standard RAM modules, there are exceptions. Most notable are laptops (and even here some standardization is starting to take hold) and high-end servers. However, even in these instances, it is likely that you will be able to find third-party RAM modules, assuming the system is not a completely new design and is relatively popular.

4.2.4. Hard Drives

All the technologies that have been discussed so far are *volatile* in nature. In other words, data contained in volatile storage technologies will be lost when the power is turned off.

Hard drives, on the other hand, are *non-volatile* — the data they contain will remain there, even after the power is removed. Because of this, hard drives occupy a special place in the storage spectrum. Their non-volatile nature makes them ideal for storing programs and data for longer-term use. Another unique aspect to hard drives is that, unlike RAM and cache memory, it is not possible to execute programs directly when they are stored on hard drives; they must first be read into RAM.

Also different from cache and RAM is the speed of data storage and retrieval; hard drives are at least an order of magnitude slower than the all-electronic technologies used for cache and RAM. The difference in speed is due mainly to their electromechanical nature. Here are the four distinct phases during data transfer to/from a hard drive. The times shown reflect how long it would take a typical high-performance drive, on average, to complete each phase:

- Access arm movement (5.5 milliseconds)
- Disk rotation (.1 milliseconds)
- Heads reading/writing data (.00014 milliseconds)
- Data transfer to/from the drive's electronics (.003 Milliseconds)

Of these, only the last phase is not dependent on any mechanical operation.


Note

Although there is much more to learn about hard drives, we will discuss disk storage technologies in more depth in Chapter 5. For the time being, it is only necessary to realize the huge speed difference between RAM and disk-based technologies and that their storage capacity usually exceeds that of RAM by a factor of at least 10, and often by 100 or more.

4.2.5. Off-Line Backup Storage

Off-line backup storage takes a step beyond hard drive storage in terms of capacity (higher) and speed (slower). Here, capacities are effectively limited only by your ability to store the removable media.

The actual technologies used in these devices can vary widely. Here are the more popular:

- Magnetic tape
- Optical disk

Of course, having removable media means that access times become even longer, particularly when the desired data is on media that is not currently in the storage device. This situation is alleviated somewhat by the use of robotic devices to automatically load and unload media, but the media storage capacities of such devices are finite, and even in the best of cases access times are measured in seconds — a far cry even from the slow multi-millisecond access times for a high-performance hard drive.

Now that we have briefly studied the various storage technologies in use today, let us explore basic virtual memory concepts.

4.3. Basic Virtual Memory Concepts

While the technology behind the construction of the various modern-day storage technologies is truly impressive, the average system administrator does not need to be aware of the details. In fact, there is really only one fact that system administrators should always keep in mind:

There is never enough RAM.

While this truism might at first seem humorous, many operating system designers have spent a great deal of time trying to reduce the impact of this very real shortage. They have done so by implementing *virtual memory* — a way of combining RAM with slower storage to give the system the appearance of having more RAM than is actually installed.

4.3.1. Virtual Memory in Simple Terms

Let us start with a hypothetical application. The machine code making up this application is 10000 bytes in size. It also requires another 5000 bytes for data storage and I/O buffers. This means that, in order to run this application, there must be 15000 bytes of RAM available; even one byte less, and the application will not be able to run.

This 15000 byte requirement is known as the application's *address space*. It is the number of unique addresses needed to hold both the application and its data. In the first computers, the amount of available RAM had to be greater than the address space of the largest application to be run; otherwise, the application would fail with an "out of memory" error.

A later approach known as *overlaying* attempted to alleviate the problem by allowing programmers to dictate which parts of their application needed to be memory-resident at any given time. In this way, code that was only required once for initialization purposes could be overlayed with code that would be used later. While overlays did ease memory shortages, it was a very complex and error-prone process. Overlays also failed to address the issue of system-wide memory shortages at runtime. In other words, an overlayed program may require less memory to run than a program that is not overlayed, but if the system still does not have sufficient memory for the overlayed program, the end result is the same — an out of memory error.

Virtual memory turns the concept of an application's address space on its head. Rather than concentrating on how *much* memory an application needs to run, a virtual memory operating system continually attempts to find the answer to the question, "how *little* memory does an application need to run?"

While it at first appears that our hypothetical application requires the full 15000 bytes to run, think back to our discussion in Section 4.1 — memory access tends to be sequential and localized. Because of this, the amount of memory required to execute the application at any given time is less than 15000 bytes — usually a lot less. Consider the types of memory accesses that would be required to execute a single machine instruction:

- The instruction is read from memory.
- The data on which the instruction will operate is read from memory.
- After the instruction completes, the results of the instruction are written back to memory.

While the actual number of bytes necessary for each memory access will vary according to the CPU's architecture, the actual instruction, and the data type, even if that one instruction required 100 bytes of memory for each type of memory access, then 300 bytes is still a lot less than the application's 15000-byte address space. If a way could be found to keep track of an application's memory requirements as the application runs, it would be possible to keep the application running using less than its address space.

But that leaves one question:

If only part of the application is in memory, where is the rest of it?

4.3.2. Backing Store — the Central Tenet of Virtual Memory

The short answer to this question is that the rest of the application remains on disk. This might at first seem to be a very large performance problem in the making — after all, are not disk drives so much slower than RAM?

While this is true, it is possible to take advantage of the sequential and localized access behavior of applications and to structure the virtual memory subsystem so that it attempts to ensure that those parts of the application that are currently needed — or likely to be needed in the near future — are kept in RAM for as long as they are needed.

In many respects this is similar to the relationship between cache and RAM: a relatively small about of fast storage can be combined with a larger amount of slow storage to look like a larger amount of fast storage.

With this in mind, let us look at the process in more detail.

4.4. Virtual Memory: the Details

First, we should introduce a new concept: *virtual address space*. As the term implies, the virtual address space is the program's address space — how much memory the program would require if it needed all the memory at once. But there is an important distinction; the word "virtual" means that this is the total number of uniquely-addressable memory locations required by the application, and *not* the amount of physical memory that must be dedicated to the application at any given time.

In the case of our example application, its virtual address space is 15000 bytes.

In order to implement virtual memory, it is necessary for the computer system to have special memory management hardware. This hardware is often known as an *MMU* (Memory Management Unit). Without an MMU, when the CPU accesses RAM, the actual RAM locations never change — memory address 123 is always the same physical location within RAM.

However, with an MMU, memory addresses go through a translation step prior to each memory access. For example, this means that memory address 123 might be directed to physical address 82043. As it turns out, the overhead of individually tracking the virtual to physical translations for billions of bytes of memory would be too much. Instead, the MMU divides RAM into *pages* — contiguous sections of memory that are handled by the MMU as single entities.

Tip

Each operating system has its own page size; in Linux (for the x86 architecture), each page is 4096 bytes long.

Keeping track of these pages and their address translations might sound like an unnecessary and confusing additional step, but it is, in fact, crucial to implementing virtual memory. For the reason why, consider the following point.

Taking our hypothetical application with the 15000 byte virtual address space, assume that the application's first instruction accesses data stored at address 12374. However, also assume that our computer only has 12288 bytes of physical RAM. What happens when the CPU attempts to access address 12374?

What happens is known as a *page fault*. Next, let us see what happens during a page fault.

4.4.1. Page Faults

First, the CPU presents the desired address (12374) to the MMU. However, the MMU has no translation for this address. So, it interrupts the CPU, and causes software known as a page fault handler to be executed. The page fault handler then determines what must be done to resolve this page fault. It can:

- Find where the desired page resides on disk and read it in (this is normally the case if the page fault is for a page of code)
- Determine that the desired page is already in RAM (but not allocated to the current process) and direct the MMU to point to it
- Point to a special page containing nothing but zeros and later allocate a page only if the page is ever written to (this is called a *copy on write* page)
- Get it from somewhere else (more on this later)

While the first three actions are relatively straightforward, the last one is not. For that, we need to cover some additional topics.

4.4.2. The Working Set, Active List and Inactive List

The group of physical memory pages currently dedicated to a specific process is known as the *working set* for that process. The number of pages in the working set can grow and shrink, depending on the overall availability of pages on a system-wide basis.

The Linux kernel keeps a list of all the pages that are actively being used. This list is known as the *active list*. As pages become less actively used, they eventually move to another list known as the *inactive list*.

The working set will grow as a process page faults (and those page faults are handled and added to the active list). The working set will shrink as fewer and fewer free pages exist — pages on the inactive list are removed from the process's working set. The operating system will shrink processes' working sets by:

- Writing modified pages to the system swap space and putting the page in the swap cache[2]
- Marking unmodified pages as being available (there is no need to write these pages out to disk as they have not changed)

In other words, the Linux memory management subsystem selects the least-recently used pages (via the inactive list) to be removed from process working sets.

4.4.3. Swapping

While swapping (writing modified pages out to the system swap space) is a normal part of a Red Hat Linux system's operation, it is possible for a system to experience too much swapping. The reason to be wary of excessive swapping is that the following situation can easily occur, over and over again:

- Pages from a process are swapped
- The process becomes runnable and attempts to access a swapped page
- The page is faulted back into memory
- A short time later, the page is swapped out again

If this sequence of events is widespread, it is known as *thrashing* and is normally indicative of insufficient RAM for the present workload. Thrashing is extremely detrimental to system perform, as the CPU and I/O loads that can be generated in such a situation can quickly outweigh the load imposed by system's real work. In extreme cases, the system may actually do no useful work, spending all its resources on moving pages to and from memory.

4.5. Virtual Memory Performance Implications

While virtual memory makes it possible for computers to more easily handle larger and more complex applications, as with any powerful tool, it comes at a price. The price in this case is one of performance — a virtual memory operating system has a lot more to do than an operating system that is not capable of virtual memory. This means that performance will never be as good with virtual memory than with the same application that is 100% memory-resident.

However, this is no reason to throw up one's hands and give up. The benefits of virtual memory are too great to do that. And, with a bit of effort, good performance is possible. The thing that must be done is to look at the system resources that are impacted by heavy use of the virtual memory subsystem.

4.5.1. Worst Case Performance Scenario

For a moment, take what you have read in this chapter, and consider what system resources are used by extremely heavy page fault and swapping activity:

- RAM — It stands to reason that available RAM will be low (otherwise there would be no need to page fault or swap).
- Disk — While disk space would not be impacted, I/O bandwidth would be.
- CPU — The CPU will be expending cycles doing the necessary processing to support memory management and setting up the necessary I/O operations for paging and swapping.

2. Under Red Hat Linux the system swap space is normally a dedicated swap partition, though swap files can also be configured and used.

The interrelated nature of these loads makes it easy to see how resource shortages can lead to severe performance problems. All it takes is:

- A system with too little RAM
- Heavy page fault activity
- A system running near its limit in terms of CPU or disk I/O

At this point, the system will be thrashing, with performance rapidly decreasing.

4.5.2. Best Case Performance Scenario

At best, system performance will present a minimal additional load to a well-configured system:

- RAM — Sufficient RAM for all working sets with enough left over to handle any page faults[3]
- Disk — Because of the limited page fault activity, disk I/O bandwidth would be minimally impacted
- CPU — The majority of CPU cycles will be dedicated to actually running applications, instead of memory management

From this, the overall point to keep in mind is that the performance impact of virtual memory is minimal when it is used as little as possible. This means that the primary determinant of good virtual memory subsystem performance is having enough RAM.

Next in line (but much lower in relative importance) are sufficient disk I/O and CPU capacity. However, these resources only help the system performance degrade more gracefully from have faulting and swapping; they do little to help the virtual memory subsystem performance (although they obviously can play a major role in overall system performance).

3. A reasonably active system will *always* experience some page faults, if for no other reason than because a newly-launched application will experience page faults as it is brought into memory.

Chapter 5.

Managing Storage

If there is one thing that takes up the majority of a system administrator's day, it would have to be storage management. It seems that disks are always running out of free space, becoming overloaded with too much I/O activity, or failing unexpectedly. Therefore, it is vital to have a solid working knowledge of disk storage in order to be a successful system administrator.

To start, let us see how disk devices are named under Red Hat Linux.

5.1. Device Naming Conventions

As with all Linux-like operating systems, Red Hat Linux uses device files to access all hardware (including disk drives). However, most of these operating systems use slightly different naming conventions to identify any attached storage devices. Here is how these device files are named under Red Hat Linux.

5.1.1. Device Files

Under Red Hat Linux, the device files for disk drives appear in the `/dev/` directory. The format for each file name depends on several aspects of the actual hardware, and how it has been configured. Here are these aspects:

- Device type
- Unit
- Partition

We will now explore each of these aspects in more detail.

5.1.1.1. Device Type

The first two letters of the device file name refer to the specific type of device. For disk drives, there are two device types that are most common:

- `sd` — The device is SCSI-based
- `hd` — The device is IDE-based

SCSI and IDE are two different industry standards that define methods for attaching devices to a computer system. The following sections briefly describe the characteristics of these two different connection technologies.

5.1.1.1.1. SCSI

Formally known as the Small Computer System Interface, the SCSI standard defines a bus along which multiple devices may be connected. A SCSI bus is a parallel bus, meaning that there is a single set of parallel wires that go from device to device. Because these wires are shared by all devices, it is necessary to have a way of uniquely identifying and communicating with an individual device. This is done by assigning each device on a SCSI bus a unique numeric address or *SCSI ID*.

Important

The number of devices that are supported on a SCSI bus depends on the width of the bus. Regular SCSI supports 8 uniquely-addressed devices, while *wide SCSI* supports 16. In either case, you must make sure that all devices are set to use a unique SCSI ID. Two devices sharing a single ID will cause problems that could lead to data corruption before it can be resolved.

One other thing to keep in mind is that *every* device on the bus uses an ID. *This includes the SCSI controller.* Quite often system administrators forget this, and unwittingly set a device to use the same SCSI ID as the bus's controller. This also means that, in practice, only 7 (or 15, for wide SCSI) devices may be present on a single bus, as each bus must include its own controller.

As technological advances have taken place, the SCSI standard has been amended to support them. For instance, the number of wires that carried data along the bus went from 8 (known simply as SCSI) to 16 (known as wide SCSI). As it became possible to build faster hardware, and the speed at which data could be transferred increased, the bus speed increased from 5MB/sec to as much as 160MB/sec. The different bus speeds are identified by adding words like "fast", "ultra", and "ultra-3" to the name of the SCSI environment being supported.

Because of SCSI's bus-oriented architecture, it is necessary to properly *terminate* both ends of the bus. Termination is accomplished by placing a load of the correct impedance on each conductor comprising the SCSI bus. Termination is an electrical requirement; without it, the various signals present on the bus would be reflected off the ends of the bus, garbling all communication.

Many (but not all) SCSI devices come with internal terminators that can be enabled or disabled using jumpers or switches. External terminators are also available.

5.1.1.1.2. IDE

IDE stands for Integrated Drive Electronics. A later version of the standard — known as *EIDE* (the extra "E" standing for "Enhanced") has been almost universally adopted in place of IDE. However, in normal conversation both are known as IDE. Like SCSI, IDE is an interface standard used to connect devices to computer systems. Like SCSI, IDE implements a bus topology.

However, there are differences between the two standards. The most important is that IDE cannot match SCSI's expandability, with each IDE bus supporting only two devices (known as a *master* and a *slave*).

5.1.1.2. Unit

Following the two-letter device type (`sd`, for example) are one or two letters denoting the specific unit. The unit designator starts with "a" for the first unit, "b" for the second, and so on. Therefore, the first hard drive on your system may appear as `hda` or `sda`.

Tip

SCSI's ability to address large numbers of devices necessitated the addition of a second unit character to support systems with more than 26 SCSI devices attached. Therefore, the first 26 SCSI hard drives would be named `sda` through `sdz`, with the 27th named `sdaa`, the 28th named `sdab`, and so on through to `sddx`.

5.1.1.3. Partition

The final part of the device file name is a number representing a specific partition on the device, starting with "1". The number may be one or two digits in length, depending on the number of partitions written to the specific device.

Once the format for device file names is known, it is easy to understand what each refers to:

- `/dev/hda1` — The first partition on the first IDE drive
- `/dev/sdb12` — The twelfth partition on the second SCSI drive
- `/dev/sdad4` — The fourth partition on the thirtieth SCSI drive

5.1.1.4. Whole-Device Access

There are instances where it is necessary to access the entire device, and not just a specific partition. This is normally done when the device is not partitioned, or does not support standard partitions (such as a CD-ROM drive). In these cases, the partition number is omitted:

- `/dev/hdc` — The entire third IDE device
- `/dev/sdb` — The entire second SCSI device

However, most disk drives use partitions; the next section will take a closer look at this method of storage division.

5.2. Partitions

Partitions are a way of dividing a disk drive's storage into distinctly separate regions. Using partitions gives the system administrator much more flexibility in terms of allocating storage.

Because they are separate from each other, partitions can have different amounts of space utilized, and that space will in no way impact the space utilized by other partitions. For example, the partition holding the files comprising the operating system will not be affected even if the partition holding the users' files becomes full. The operating system will still have free space for its own use.

Although it is somewhat simplistic, from this perspective you can think of partitions as being similar to individual disk drives. In fact, some operating systems actually refer to partitions as "drives". However, this viewpoint is not entirely accurate; therefore, it is important that we look at partitions more closely.

5.2.1. Partition Attributes

Partitions are defined by the following attributes:

- Partition geometry
- Partition type
- Partition type field

Next, we will explore these attributes in more detail.

5.2.1.1. Geometry

A partition's geometry refers to its physical placement on a disk drive. In order to understand geometry, we must first understand how data is stored on a disk drive.

As the name implies, a disk drive contain one or more disks coated with a magnetic material. It is this material that actually stores the data. The surface of each disk is read and written by a *head*, similar in function to the head in a cassette tape recorder.

The head for each disk surface is attached to an *access arm*, which allows the heads to sweep across the surfaces of the disks. As the disks rotate under the heads, the section of the disks under the heads at any given position of the access arm make up a *cylinder* (when only one disk surface is involved, this circular slice of magnetic media is known as a *track*). Each track making up each cylinder is further divided into *sectors*; these fixed-sized pieces of storage represent the smallest directly-addressable items on a disk drive. There are normally hundreds of sectors per track. Present-day disk drives may have tens of thousands of cylinders, representing tens of thousands of unique positions of the access arm.

Partitions are normally specified in terms of cylinders, with the partition size is defined as the amount of storage between the starting and ending cylinders.

5.2.1.2. Partition Type

The partition type refers to the partition's relationship with the other partitions on the disk drive. There are three different partition types:

- Primary partitions
- Extended partitions
- Logical partitions

We will now look at each partition type.

5.2.1.2.1. Primary Partitions

Primary partitions are partitions that take up one of the four primary partition slots in the disk drive's partition table.

5.2.1.2.2. Extended Partitions

Extended partitions were developed in response to the need for more than four partitions per disk drive. An extended partition can itself contain multiple partitions, greatly extending the number of partitions possible.

5.2.1.2.3. Logical Partitions

Logical partitions are those partitions contained within an extended partition.

5.2.1.3. Partition Type Field

Each partition has a type field that contains a code indicating the partition's anticipated usage. In other words, if the partition is going to be used as a swap partition under Red Hat Linux, the partition's type should be set to 82 (which is the code representing a Linux swap partition).

5.3. File System Basics

A disk drive by itself provides a place to store data, and nothing more. In fact, by itself, the only way to access data on a hard drive is by either specifying the data's physical location (in terms of cylinder, head, and sector), or by its logical location (the 65,321st block) on the disk.

What is needed is a way to more easily keep track of things stored on hard drives; a way of filing information in an easily-accessible way.

That is the role of the *file system*.

5.3.1. An Overview of File Systems

File systems, as the name implies, treat different sets of information as files. Each file is separate from every other. Over and above the information stored within it, each file includes additional information:

- The file's name
- The file's access permissions
- The time and date of the file's creation, access, and modification.

While file systems in the past have included no more complexity than that already mentioned, present-day file systems include mechanisms to make it easier to group related files together. The most commonly-used mechanism is the directory. Often implemented as a special type of file, directories make it possible to create hierarchical structures of files and directories.

However, while most file systems have these attributes in common, they vary in implementation details, meaning that not all file systems can be accessed by all operating systems. Luckily, Red Hat Linux includes support for many popular file systems, making it possible to easily access the file systems of other operating systems.

This is particularly useful in dual-boot scenarios, and when migrating files from one operating system to another.

Next, we will examine some of file systems that are frequently used under Red Hat Linux.

5.3.1.1. EXT2

Until recently, the ext2 file system has been the standard Linux file system for Red Hat Linux. As such, it has received extensive testing, and is considered one of the more robust file systems in use today.

However, there is no perfect file system, and ext2 is no exception. One problem that is very commonly reported is that an ext2 file system must undergo a lengthy file system integrity check if the system was not cleanly shut down. While this requirement is not unique to ext2, the popularity of ext2, combined with the advent of larger disk drives, meant that file system integrity checks were taking longer and longer. Something had to be done.

5.3.1.2. EXT3

The ext3 file system builds upon ext2 by adding journaling capabilities to the already-proven ext2 codebase. As a journaling file system, ext3 always keeps the file system in a consistent state, eliminating the need for file system integrity checks.

This is accomplished by writing all file system changes to an on-disk journal, which is then flushed on a regular basis. After an unexpected system event (such as a power outage or system crash), the only operation that needs to take place prior to making the file system available is to process the contents of the journal; in most cases this takes approximately one second.

Because ext3's on-disk data format is based on ext2, it is possible to access an ext3 file system on any system capable of reading and writing an ext2 file system (without the benefit of journaling, however). This can be a sizable benefit in organizations where some systems are using ext3 and some are still using ext2.

5.3.1.3. NFS

As the name implies, the Network File System (more commonly known as NFS) is a file system that may be accessed via a network connection. With other file systems, the storage device must be directly attached to the local system. However, with NFS this is not a requirement, making possible a variety of different configurations, from centralized file system servers, to entirely diskless computer systems.

However, unlike the other file systems discussed here, NFS does not dictate a specific on-disk format. Instead, it relies on the server operating system's native file system support to control the actual I/O to local disk drive(s). NFS then makes the file system available to any operating system running a compatible NFS client.

While primarily a Linux and UNIX technology, it is worth noting that NFS client implementations exist for other operating systems, making NFS a viable technique to share files with a variety of different platforms.

5.3.1.4. ISO 9660

In 1987, the International Organization for Standardization (known as ISO) released international standard 9660. ISO 9660 defines how files are represented on CD-ROMs. Red Hat Linux system administrators will likely see ISO 9660-formatted data in two places:

- CD-ROMs
- Files containing complete ISO 9660 file systems, meant to be written to CD-R or CD-RW media

The basic ISO 9660 standard is rather limited in functionality, especially when compared with more modern file systems. File names may be a maximum of eight characters long and an extension of no more than three characters is permitted (often known as `8.3` file names). However, various extensions to the standard have become popular over the years, among them:

- Rock Ridge — Uses some fields undefined in ISO 9660 to provide support features such as long mixed-case file names, symbolic links, and nested directories (in other words, directories that can themselves contain other directories)
- Joliet — An extension of the ISO 9660 standard, developed by Microsoft to allow CD-ROMs to contain long file names, using the Unicode character set

Red Hat Linux is able to correctly interpret ISO 9660 file systems using both the Rock Ridge and Joliet extensions.

5.3.1.5. MSDOS

Red Hat Linux also supports file systems from other operating systems. As the name for the msdos file system implies, the original operating system was Microsoft's MS-DOS®. As in MS-DOS, a Red Hat Linux system accessing an msdos file system is limited to 8.3 file names. Likewise, other file attributes such as permissions and ownership cannot be changed. However, from a file interchange standpoint, the msdos file system is more than sufficient to get the job done.

5.3.1.6. VFAT

The vfat file system was first used by Microsoft's Windows® series of operating systems. An improvement over the msdos file system, file names on a vfat file system may be longer than msdos's 8.3. However, permissions and ownership still cannot be changed.

Now that we have seen which file systems are most commonly used under Red Hat Linux, let us see how they are used.

5.3.2. Mounting File Systems

In order to access any file system, it is first necessary to *mount* it. By mounting a file system, you direct Red Hat Linux to make a specific device (and partition) available to the system. Likewise, when access to a particular file system is no longer desired, it is necessary to *umount* it.

In order to mount any file system, two pieces of information must be specified:

- A device file representing the desired disk drive and partition
- A directory under which the mounted file system will be made available (otherwise known as a *mount point*

We have already covered the device files earlier (in Section 5.1), so the following section will discuss mount points in more detail.

5.3.2.1. Mount Points

Unless you are used to Linux (or Linux-like) operating systems, the concept of a mount point will at first seem strange. However, it is one of the most powerful methods of managing files ever developed. With many other operating systems, a full file specification includes the file name, some means of identifying the specific directory in which the file resides, and a means of identifying the physical device on which the file can be found.

With Red Hat Linux, a slightly different approach is used. As with other operating systems, a full file specification includes the file's name and the directory in which it resides. However, there is no explicit device specifier.

The reason for this apparent shortcoming is the mount point. On other operating systems, there is one directory hierarchy for each partition. However, on Linux-like systems, there is only *one* hierarchy system-wide and this single directory hierarchy can span multiple partitions. The key is the mount point. When a file system is mounted, that file system is made available as a set of subdirectories under the specified mount point.

This apparent shortcoming is actually a strength. It means that seamless expansion of a Linux file system is possible, with every directory capable of acting as a mount point for additional disk space.

As an example, assume a Red Hat Linux system contained a directory `foo` in its root directory; the full path to the directory would be `/foo`. Next, assume that this system has a partition that is to be mounted, and that the partition's mount point is to be `/foo`. If that partition had a file by the name of `bar.txt` in its top-level directory, after the partition was mounted you could access the file with the following full file specification:

```
/foo/bar.txt
```

In other words, once this partition has been mounted, any file that is read or written anywhere under the `/foo` directory will be read from or written to the partition.

A commonly-used mount point on many Red Hat Linux systems is `/home` — that is because all user accounts' login directories normally are located under `/home`, meaning that all users' files can be written to a dedicated partition, and not fill up the operating system's file system.


Tip

Since a mount point is just an ordinary directory, it is possible to write files into a directory that is later used as a mount point. If this happens, what happens to the files that were in the directory originally?

For as long as a partition is mounted on the directory, the files are not accessible. However, they will not be harmed, and can be accessed after the partition is unmounted.

5.3.2.2. Seeing What is Mounted

In addition to mounting and unmounting disk space, it is possible to see what is mounted. There are several different ways of doing this:

- Viewing `/etc/mtab`
- Viewing `/proc/mounts`
- Issuing the `df` command

5.3.2.2.1. Viewing `/etc/mtab`

The file `/etc/mtab` is a normal file that is updated by the `mount` program whenever file systems are mounted or unmounted. Here is a sample `/etc/mtab`:

```
/dev/sda3 / ext3 rw 0 0
none /proc proc rw 0 0
usbdevfs /proc/bus/usb usbdevfs rw 0 0
/dev/sda1 /boot ext3 rw 0 0
none /dev/pts devpts rw,gid=5,mode=620 0 0
/dev/sda4 /home ext3 rw 0 0
none /dev/shm tmpfs rw 0 0
automount(pid1006) /misc autofs rw,fd=5,pgrp=1006,minproto=2,maxproto=3 0 0
none /proc/sys/fs/binfmt_misc binfmt_misc rw 0 0
```

Each line represents a file system that is currently mounted and contains the following fields (from left to right):

- The device specification
- The mount point
- The file system type
- Whether the file system is mounted read-only (`ro`) or read-write (`rw`), along with any other mount options
- Two unused fields with zeros in them (for compatibility with `/etc/fstab`)

5.3.2.2.2. Viewing `/proc/mounts`

The `/proc/mounts` file is part of the proc virtual file system. As with the other files under `/proc/`, `mounts` does not exist on any disk drive in your Red Hat Linux system. Instead, these files are representations of system status made available in file form. Using the command `cat /proc/mounts`, we can view `/proc/mounts`:

```
rootfs / rootfs rw 0 0
/dev/root / ext3 rw 0 0
/proc /proc proc rw 0 0
usbdevfs /proc/bus/usb usbdevfs rw 0 0
/dev/sda1 /boot ext3 rw 0 0
none /dev/pts devpts rw 0 0
/dev/sda4 /home ext3 rw 0 0
none /dev/shm tmpfs rw 0 0
none /proc/sys/fs/binfmt_misc binfmt_misc rw 0 0
```

As we can see from the above example, the format of `/proc/mounts` is very similar to that of `/etc/mtab`. There are a number of file systems mounted that have nothing to do with disk drives. Among these are the `/proc/` file system itself (along with two other file systems mounted under `/proc/`), pseudo-ttys, and shared memory.

While the format is admittedly not very user-friendly, looking at `/proc/mounts` is the best way to be 100% sure of seeing what is mounted on your Red Hat Linux system. Other methods can, under rare circumstances, be inaccurate.

However, most of the time you will likely use a command with more easily-read (and useful) output. Let us look at that command next.

5.3.2.2.3. The `df` Command

While using `/proc/mounts` will let you know what file systems are currently mounted, it does little beyond that. Most of the time you will be more interested in one particular aspect of the file systems that are currently mounted:

The amount of free space on them.

For this, we can use the `df` command. Here is some sample output from `df`:

```
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/sda3              8428196   4280980   3719084  54% /
/dev/sda1               124427     18815     99188  16% /boot
/dev/sda4              8428196   4094232   3905832  52% /home
none                    644600         0    644600   0% /dev/shm
```

Several differences with `/etc/mtab` and `/proc/mount` are immediately obvious:

- An easy-to-read heading is displayed
- With the exception of the shared memory file system, only disk-based file systems are shown
- Total size, used space, free space, and percentage in use figures are displayed

That last point is probably the most important, because every system administrator will eventually have to deal with a system that has run out of free disk space. With `df` it is very easy to see where the problem lies.

5.3.3. Mounting File Systems Automatically with `/etc/fstab`

When a Red Hat Linux system is newly-installed, all the disk partitions defined and/or created during the installation are configured to be automatically mounted whenever the system boots. However, what happens when additional disk drives are added to a system after the installation is done? The answer is "nothing" because the system was not configured to mount them automatically. However, this is easily changed.

The answer lies in the `/etc/fstab` file. This file is used to control what systems are mounted when the system boots, as well as to supply default values for other file systems that may be mounted manually from time to time. Here is a sample `/etc/fstab` file:

```
LABEL=/               /                   ext3    defaults         1 1
LABEL=/boot           /boot               ext3    defaults         1 2
none                  /dev/pts            devpts  gid=5,mode=620   0 0
LABEL=/home           /home               ext3    defaults         1 2
none                  /proc               proc    defaults         0 0
none                  /dev/shm            tmpfs   defaults         0 0
/dev/sda2             swap                swap    defaults         0 0
/dev/cdrom            /mnt/cdrom          iso9660 noauto,owner,kudzu,ro 0 0
/dev/fd0              /mnt/floppy         auto    noauto,owner,kudzu 0 0
```

Each line represents one file system, and contains the following fields:

- File system specifier — For disk-based file systems, either a device file, or a device label specification
- Mount point — Except swap partitions, this field specifies the mount point to be used when the file system is mounted
- File system type — The type of file system present on the specified device (note that `auto` may be specified to select automatic detection of the file system to be mounted, which is handy for CD-ROMs and diskette drives)
- Mount options — A comma-separated list of options that can be used to control `mount`'s behavior
- Dump frequency — If the `dump` backup utility is used, the number in this field will control `dump`'s handling of the specified file system
- File system check order — Controls the order in which the file system checker `fsck` checks the integrity of the file systems.

5.4. Adding/Removing Storage

Because the need for additional disk space is never-ending, a system administrator often will need to add disk space, while often removing older, smaller drives. In this section, we will go over the basic process of adding and removing storage on a Red Hat Linux system.

5.4.1. Adding Storage

The process of adding storage to a Red Hat Linux system is relatively straightforward. Here are the basic steps:

1. Installing the hardware
2. Partitioning
3. Formatting the partition(s)

4. Updating `/etc/fstab`
5. Modifying backup schedule

Let us look at each step in more detail.

5.4.1.1. Installing the Hardware

Before anything else can be done, the new disk drive has to be in place and accessible. While there are many different hardware configurations possible, we will go through the two most common situations — adding an IDE or SCSI disk drive. Even with other configurations, the basic steps outlined here still apply.


Tip

No matter what storage hardware you use, you should always consider the load a new disk drive will add to your computer's I/O subsystem. In particular, you should try to spread the disk I/O load over all available channels/buses. From a performance standpoint, this is far better than putting all disk drives on one channel and leaving another one empty and idle.

5.4.1.1.1. Adding IDE Disk Drives

IDE disk drives are mostly used in desktop and lower-end server systems. Nearly all systems in these classes have built-in IDE controllers with multiple IDE channels — normally two or four.

Each channel can support two devices — one master, and one slave. The two devices are connected to the channel with a single cable. Therefore, the first step is to see which channels have available space for an addition disk drive. You will find one of three situations:

- There is a channel with only one disk drive connected to it
- There is a channel with no disk drive connected to it
- There is no space available

The first situation is usually the easiest, as it is very likely that the cable in place has an unused connector into which the new disk drive can be plugged. However, if the cable in place only has two connectors (one for the channel and one for the already-installed disk drive), then it will be necessary to replace the existing cable with a three-connector model.

Before installing the new disk drive, make sure that the two disk drives sharing the channel are appropriately configured (one as master and one as slave).

The second situation is a bit more difficult, if only for the reason that a cable must be purchased in order to connect a disk drive to the channel. The new disk drive may be configured as master or slave (although traditionally the first disk drive on a channel is normally configured as master).

In the third situation, there is no space left for an additional disk drive. You must then make a decision. Do you:

- Acquire an IDE controller card, and install it
- Replace one of the installed disk drives with the newer, larger one

Adding a controller card entails checking hardware compatibility, physical capacity, and software compatibility. Basically, the card must be compatible with your computer's bus slots, there must be an open slot for it, and it must be supported by Red Hat Linux.

Replacing an installed disk drive presents a unique problem: what to do with the data on the disk? There are a few possible approaches:

- Write the data to a backup device and restore after installing the new disk drive
- Use your network to copy the data to another system with sufficient free space, restoring the data after installing the new disk drive
- Use the space occupied by a third disk drive by:
    1. Temporarily removing some other disk drive
    2. Temporarily installing the new disk drive in its place
    3. Copying the data to the new disk drive
    4. Removing the old disk drive
    5. Replacing it with the new disk drive
    6. Reinstalling the temporarily removed disk drive
- Temporarily install the original disk drive and the new disk drive in another computer, copy the data to the new disk drive, and then install the new disk drive in the original computer

As you can see, sometimes a fair bit of effort must be expended to get the data (and the new hardware) where it needs to go. Next, we will look at working with SCSI disk drives.

5.4.1.1.2. Adding SCSI Disk Drives

SCSI disk drives normally are used in higher-end workstations and server systems. Unlike IDE-based systems, SCSI systems may or may not have built-in SCSI controllers; some do, while others use a separate SCSI controller card.

The capabilities of SCSI controllers (whether built-in or not) also vary widely. It may supply a narrow or wide SCSI bus. The bus speed may be normal, fast, ultra, utra2, or ultra160.

If these terms are unfamiliar to you, you will have to determine which term applies to your hardware configuration and select an appropriate new disk drive. The best resource for this information would be the documentation for your system and/or SCSI adapter.

You must then determine how many SCSI buses are available on your system, and which ones have available space for a new disk drive. The number of devices supported by a SCSI bus will vary according to the bus width:

- Narrow (8-bit) SCSI bus — 7 devices (plus controller)
- Wide (16-bit) SCSI bus — 15 devices (plus controller)

The first step is to see which buses have available space for an additional disk drive. You will find one of three situations:

- There is a bus with less than the maximum number of disk drives connected to it
- There is a bus with no disk drives connected to it
- There is no space available on any bus

The first situation is usually the easiest, as it is likely that the cable in place has an unused connector into which the new disk drive can be plugged. However, if the cable in place does not have an unused connector, it will be necessary to replace the existing cable with one that has at least one more connector.

The second situation is a bit more difficult, if only for the reason that a cable must be purchased in order to connect a disk drive to the bus.

If there is no space left for an additional disk drive, you must make a decision. Do you:

- Acquire and install a SCSI controller card
- Replace one of the installed disk drives with the new one

Adding a controller card entails checking hardware compatibility, physical capacity, and software compatibility. Basically, the card must be compatible with your computer's bus slots, there must be an open slot for it, and it must be supported by Red Hat Linux.

Replacing an installed disk drive presents a unique problem: what to do with the data on the disk? There are a few possible approaches:

- Write the data to a backup device, and restore after installing the new disk drive
- Use your network to copy the data to another system with sufficient free space, and restore after installing the new disk drive
- Use the space occupied by a third disk drive by:
    1. Temporarily removing some other disk drive
    2. Temporarily installing the new disk drive in its place
    3. Copying the data to the new disk drive
    4. Removing the old disk drive
    5. Replacing it with the new disk drive
    6. Reinstalling the temporarily removed disk drive
- Temporarily install the original disk drive and the new disk drive in another computer, copy the data to the new disk drive, and then install the new disk drive in the original computer

Once you have an available connector in which to plug the new disk drive, you must make sure that the drive's SCSI ID is set appropriately. To do this, you must know what all of the other devices on the bus (including the controller) are using for their SCSI IDs. The easiest way to do this is to access the SCSI controller's BIOS. This is normally done by pressing a specific key sequence during the system's power-up sequence. You can then view the SCSI controller's configuration, along with the devices attached to all of its buses.

Next, you must consider proper bus termination. When adding a new disk drive, the rule is actually quite simple — if the new disk drive is the last (or only) device on the bus, it must have termination enabled. Otherwise, termination must be disabled.

At this point, you can move on to the next step in the process — partitioning your new disk drive.

5.4.1.2. Partitioning

Once the disk drive has been installed, it is time to create one or more partitions to make the space available to Red Hat Linux. There are several different ways of doing this:

- Using the command-line `fdisk` utility program
- Using `parted`, another command-line utility program

Although the tools may be different, the basic steps are the same:

1. Select the new disk drive (the drive's name can be found by following the device naming conventions outlined in Section 5.1
2. View the disk drive's partition table, to ensure that the disk drive to be partitioned is, in fact, the correct one

3. Delete any unwanted partitions that may already be present on the new disk drive
4. Create the new partition(s), being sure to specify the desired size and file system type
5. Save your changes and exit the partitioning program

Warning

When partitioning a new disk drive, it is *vital* that you are sure the disk drive you are about to partition is the correct one. Otherwise, you may inadvertently partition a disk drive that is already in use, which will result in lost data.

Also make sure you have decided on the best partition size. Always give this matter serious thought, because changing it later will be much more difficult.

5.4.1.3. Formatting the Partition(s)

At this point, the new disk drive has one or more partitions that have been written to it. However, before the space contained within those partitions can be used, the disk drive must first be formatted. By formatting, you are selecting a specific file system to be used — this is the step that turns that blank space into an EXT3 file system, for example. As such, this is a pivotal time in the life of this disk drive; the choices you make here cannot be changed later without going through a great deal of work.

This is the time to look at the `mkfs.<fstype>` man page for the file system you have selected. For example, look at the `mkfs.ext3` man page to see the options available to you when creating a new ext3 file system. In general, the `mkfs.*` programs provide reasonable defaults for most configurations; however here are some of the options that system administrators most commonly change:

- Setting a volume label for later use in `/etc/fstab`
- On very large hard disks, setting a lower percentage of space reserved for the super-user
- Setting a non-standard block size and/or bytes per inode for configurations that must support either very large or very small files
- Checking for back blocks before formatting

The disk drive is now properly configured for use.

Next, it is always best to double-check your work by manually mounting the partition(s) and making sure everything is in order. Once everything checks out, it is time to configure your Red Hat Linux system to automatically mount the new file system(s) whenever it boots.

5.4.1.4. Updating `/etc/fstab`

As outlined in Section 5.3.3, you must add the necessary line(s) to `/etc/fstab` in order to ensure that the new file system(s) are mounted whenever the system reboots. Once you have updated `/etc/fstab`, test your work by issuing an "incomplete" `mount`, specifying only the device or mount point. Something similar to one of the following will be sufficient:

```
mount /home
mount /dev/hda3
```

(Replacing `/home` or `/dev/hda3` with the mount point or device for your specific situation.)

If the appropriate `/etc/fstab` entry is correct, `mount` will obtain the missing information from it, and complete the file system mount.

At this point you can be relatively confident that the new file system will be there the next time the system boots (although if you can afford a quick reboot, it would not hurt to do so — just to be sure).

Next, we will look at the one of the most commonly-forgotten steps in the process of adding a new file system.

5.4.1.5. Modifying the Backup Schedule

Assuming that the new file system is more than a temporary storage area requiring no backups, this is the time to make the necessary changes to your backup procedures to ensure that the new file system will be backed up. The exact nature of what you will need to do to make this happen depends on the way that backups are performed on your system. However, there are some points to keep in mind while making the necessary changes:

- Consider what the optimal frequency of backups should be
- Determine what backup style would be most appropriate (full backups only, full with incrementals, full with differentials, etc.)
- Consider the impact of the new file system on your backup media usage, particularly as the new file system starts to fill
- Judge whether the additional backup will cause the backups to take too long and start using time outside of your backup window
- Make sure that these changes are communicated to the people that need to know (other system administrators, operations personnel, etc.)

Once all this is done, your new disk space is ready for use.

5.4.2. Removing Storage

Removing disk space from a system is straightforward, with the steps being similar to the installation sequence (except, of course, in reverse):

1. Move any data to be saved off the disk drive
2. Remove the disk drive from the backup system
3. Remove the disk drive's partitions from `/etc/fstab`
4. Erase the contents of the disk drive
5. Remove the disk drive

As you can see, compared to the installation process, there are a few extra steps here.

5.4.2.1. Moving Data Off the Disk Drive

Should there be any data on the disk drive that must be saved, the first thing to do is to determine where the data should go. The decision here depends mainly on what is going to be done with the data. For example, if the data is no longer going to be actively used, it should be archived, probably in the same manner as your system backups. This means that now is the time to consider appropriate retention periods for this final backup.

On the other hand, if the data will still be used, then the data will need to reside on the system most appropriate for that usage. Of course, if this is the case, perhaps it would be easiest to move the data by simply reinstalling the disk drive on the new system. If you do this, you should make a full backup of

the data before doing so — people have dropped disk drives full of valuable data (losing everything) while doing nothing more than walking across a room.

5.4.2.2. Erase the Contents of the Disk Drive

No matter whether the disk drive has valuable data or not, it is a good idea to always erase a disk drive's contents prior to reassigning or relinquishing control of it. While the obvious reason is to make sure that no information remains on the disk drive, it is also a good time to check the disk drive's health by performing a read-write test for bad blocks on the entire drive.

Doing this under Red Hat Linux is simple. After unmounting all of the disk drive's partitions, issue the following command (while logged in as root):

```
badblocks -ws /dev/fd0
```

You will see the following output while `badblocks` runs:

```
Writing pattern 0xaaaaaaaa: done
Reading and comparing: done
Writing pattern 0x55555555: done
Reading and comparing: done
Writing pattern 0xffffffff: done
Reading and comparing: done
Writing pattern 0x00000000: done
Reading and comparing: done
```

In this example, a diskette (`/dev/fd0`) was erased; however, erasing a hard disk is done the same way, using full-device access (for example, `/dev/hda` for the first IDE hard disk)

Important

Many companies (and government agencies) have specific methods of erasing data from disk drives and other data storage media. You should *always* be sure you understand and abide by these requirements; in many cases there are legal ramifications if you fail to do so. The example above should in no way be considered the ultimate method of wiping a disk drive.

5.5. RAID-Based Storage

One skill that a system administrator should cultivate is the ability to look at complex system configurations, and observe the different shortcomings inherent in each configuration. While this might, at first glance, seem to be a rather depressing viewpoint to take, it can be a great way to look beyond the shiny new boxes to some future Saturday night with all production down due to a failure that could easily have been avoided.

With this in mind, let us use what we now know about disk-based storage and see if we can determine the ways that disk drives can cause problems. First, consider an outright hardware failure:

A disk drive with four partitions on it dies completely: what happens to the data on those partitions? It is immediately unavailable (at least until it can be restored from a recent backup, that is).

A disk drive with a single partition on it is operating at the limits of its design due to massive I/O loads: what happens to applications that require access to the data on that partition? The applications slow down because the disk drive cannot process reads and writes any faster.

You have a large data file that is slowly growing in size; soon it will be larger than the largest disk drive available for your system. What happens then? The data file (and its associated applications) stop running.

Just one of these problems could cripple a data center, yet system administrators must face these kinds of issues every day. What can be done?

Fortunately, there is one technology that can address each one of these issues. And the name for that technology is RAID.

5.5.1. Basic Concepts

RAID is an acronym standing for Redundant Array of Independent Disks[1]. As the name implies, RAID is a way for multiple disk drives to act as a single disk drive.

RAID techniques were first developed by researchers at the University of California, Berkeley in the mid-1980s. At the time, there was a large gap in price between the high-performance disk drives used on the large computers installations of the day, and the smaller, slower disk drives used by the still-young personal computer industry. RAID was viewed as a method of having many less expensive disk drives fill in for higher-priced hardware.

More importantly, RAID arrays can be constructed in different ways, and will have different characteristics depending on the final configuration. Let us look at the different configurations (known as RAID *levels*) in more detail.

5.5.1.1. RAID Levels

The Berkeley researchers originally defined five different RAID levels and numbered them "1" through "5". In time, additional RAID levels were defined by other researchers and members of the storage industry. Not all RAID levels were equally useful; some were of interest only for research purposes, and others could not be economically implemented.

In the end, there were three RAID levels that ended up seeing widespread usage:

- Level 0
- Level 1
- Level 5

The following sections will discuss each of these levels in more detail.

5.5.1.1.1. RAID 0

The disk configuration known as RAID level 0 is a bit misleading, as this is the only RAID level that employs absolutely no redundancy. However, even though RAID 0 has no advantages from a reliability standpoint, it does have other advantages.

A RAID 0 array consists of two or more disk drives. The drives are divided into *chunks*, which represents some multiple of the drives' native block size. Data written to the array will be written, chunk by chunk, to each drive in the array. The chunks can be thought of as forming stripes across each drive in the array; hence the other term for RAID 0: *striping*.

For example, with a two-drive array and a 4KB chunk size, writing 12KB of data to the array would result in the data being written in three 4KB chunks to the following drives:

1. When early RAID research began, the acronym stood for Redundant Array of *Inexpensive* Disks, but over time the "standalone" disks that RAID was intended to supplant became cheaper and cheaper, rendering the price comparison meaningless.

- The first 4KB would be written to the first drive, into the first chunk
- The second 4KB would be written to the second drive, into the second chunk
- The last 4KB would be written to the first drive, into the second chunk

5.5.1.1.1.1. Advantages to RAID 0

Compared to a single disk drive, the advantages to RAID 0 are:

- Larger total size — RAID 0 arrays can be constructed that are larger than a single disk drive, making it easier to store larger data files
- Better read/write performance — The I/O load on a RAID 0 array will be spread evenly among all the drives in the array
- No wasted space — All available storage on all drives in the array are available for data storage

5.5.1.1.1.2. Disadvantages to RAID 0

Compared to a single disk drive, RAID 0 has the following disadvantage:

- Less reliability — Every drive in a RAID 0 array must be operative in order for the array to be available

Tip

If you have trouble keeping the different RAID levels straight, just remember that RAID 0 has *zero* percent redundancy.

5.5.1.1.2. RAID 1

RAID 1 uses two (although some implementations support more) identical disk drives. All data is written to both drives, making them identical copies of each other. That is why RAID 1 is often known as *mirroring*.

Whenever data is written to a RAID 1 array, two physical writes must take place: one to one drive, and one to the other. Reading data, on the other hand, only needs to take place once and either drive in the array can be used.

5.5.1.1.2.1. Advantages to RAID 1

Compared to a single disk drive, a RAID 1 array has the following advantages:

- Improved redundancy — Even if one drive in the array were to fail, the data would still be accessible
- Improved read performance — With both drives operational, reads can be evenly split between them

5.5.1.1.2.2. Disadvantages to RAID 1

When compared to a single disk drive, a RAID 1 array has some disadvantages:

- Reduced write performance — Because both drives must be kept up-to-date, all write I/O must be performed by both drives, slowing the overall process of writing data to the array
- Reduced cost efficiency — With one entire drive dedicated to redundancy, the cost of a RAID 1 array is at least double that of a single drive

5.5.1.1.3. RAID 5

RAID 5 attempts to combine the benefits of RAID 0 and RAID 1, while minimizing their respective disadvantages.

Like RAID 0, a RAID 5 array consists of multiple disk drives, each divided into chunks. This allows a RAID 5 array to be larger than any single drive. And like a RAID 1 array, a RAID 5 array uses some disk space in a redundant fashion, improving reliability.

However, the way RAID 5 works is unlike either RAID 0 or 1.

A RAID 5 array must consist of at least three identically-sized disk drives (although more drives may be used). Each drive is divided into chunks and data is written to the chunks in order. However, not every chunk is dedicated to data storage as it is in RAID 0. Instead, in an array with n disk drives in it, every nth chunk is dedicated to *parity*.

Chunks containing parity make it possible to recover data should one of the drives in the array fail. The parity in chunk x is calculated by mathematically combining the data from each chunk x stored on all the other drives in the array. If the data in a chunk is updated, the corresponding parity chunk must be recalculated and updated as well.

This also means that every time data is written to the array, *two* drives are written to: the drive holding the data, and the drive containing the parity chunk.

One key point to keep in mind is that the parity chunks are not concentrated on any one drive in the array. Instead, they are spread evenly through all the drives. Even though dedicating a specific drive to contain nothing but parity is possible (and, in fact, this configuration is known as RAID level 4), the constant updating of parity as data is written to the array would mean that the parity drive could become a performance bottleneck. By spreading the parity information throughout the array, this impact is reduced.

5.5.1.1.3.1. Advantages to RAID 5

Compared to a single drive, a RAID 5 array has the following advantages:

- Improved redundancy — If one drive in the array fails, the parity information can be used to reconstruct the missing data chunks, all while keeping the data available for use
- Improved read performance — Due to the RAID 0-like way data is divided between drives in the array, read I/O activity is spread evenly between all the drives
- Reasonably good cost efficiency — For a RAID 5 array of n drives, only $1/n$th of the total available storage is dedicated to redundancy

5.5.1.1.3.2. Disadvantages to RAID 5

Compared to a single drive, a RAID 5 array has the following disadvantage:

- Reduced write performance — Because each write to the array results in two writes to the physical drives (one write for the data and one for the parity), write performance is worse than a single drive[2]

5.5.1.1.4. Nested RAID Levels

As should be obvious from the discussion of the various RAID levels, each level has specific strengths and weaknesses. It was not long before people began to wonder whether different RAID levels could somehow be combined, producing arrays with all of the strengths and none of the weaknesses of the original levels.

For example, what if the disk drives in a RAID 0 array were actually RAID 1 arrays? This would give the advantages of RAID 0's speed, with the reliability of RAID 1.

This is just the kind of thing that can be done. Here are the most commonly-nested RAID levels:

- RAID 1+0
- RAID 5+0
- RAID 5+1

Because nested RAID is used in more specialized environments, we will not go into greater detail here. However, there are two points to keep in mind when thinking about nested RAID:

- Order matters — The order in which RAID levels are nested can have a large impact on reliability. In other words, RAID 1+0 and RAID 0+1 are *not* the same
- Costs can be high — If there is any disadvantage common to all nested RAID implementations, it is one of cost; the smallest possible RAID 5+1 array is six disk drives (and even more drives will be required for larger arrays)

Now that we have explored the concepts behind RAID, let us see how RAID can be implemented.

5.5.1.2. RAID Implementations

It is obvious from the previous sections that RAID requires additional "intelligence" over and above the usual disk I/O processing for individual drives. At the very least, the following tasks must be performed:

- Dividing incoming I/O requests to the individual disks in the array
- Calculating parity (for RAID 5), and writing it to the appropriate drive in the array
- Monitoring the individual disks in the array and taking the appropriate actions should one fail
- Controlling the rebuilding of an individual disk in the array, when that disk has been replaced or repaired
- Providing a means to allow administrators to maintain the array (removing and adding drives, initiating and halting rebuilds, etc.)

2. There is also an impact from the parity calculations required for each write. However, depending on the specific RAID 5 implementation, this impact can range from sizable to nearly nonexistent.

Fortunately, there are two major methods that may be used to accomplish these tasks. The next two sections will describe them.

5.5.1.2.1. Hardware RAID

A hardware RAID implementation usually takes the form of a specialized disk controller card. The card performs all RAID-related functions and directly controls the individual drives in the arrays attached directly to it. With the proper driver, the arrays managed by a hardware RAID card appear to the host operating system just as if they were regular disk drives.

Most RAID controller cards work with SCSI drives, although there are some IDE-based RAID controllers as well. In any case, the administrative interface is usually implemented in one of three ways:

- Specialized utility programs that run as applications under the host operating system
- An on-board interface using a serial port that is accessed using a terminal emulator
- A BIOS-like interface that is only accessible during the system's power-up testing

Some RAID controllers have more than one type of administrative interface available. For obvious reasons, a software interface provides the most flexibility, as it allows administrative functions while the operating system is running. However, if you are going to boot Red Hat Linux from a RAID controller, an interface that does not require a running operating system is a requirement.

Because there are so many different RAID controller cards on the market, it is impossible to go into further detail here. The best course of action is to read the manufacturer's documentation for more information.

5.5.1.2.2. Software RAID

Software RAID is simply RAID implemented as kernel- or driver-level software for a particular operating system. As such, it provides more flexibility in terms of hardware support — as long as the hardware is supported by the operating system, RAID arrays can be configured and deployed. This can dramatically reduce the cost of deploying RAID by eliminating the need for expensive, specialized RAID hardware.

Because Red Hat Linux includes support for software RAID, the remainder of this section will describe how it may be configured and deployed.

5.5.2. Creating RAID Arrays

Under Red Hat Linux there are two ways that RAID arrays can be created:

- While installing Red Hat Linux
- Manually, after Red Hat Linux has been installed

We will next look into these two methods.

5.5.2.1. While Installing Red Hat Linux

During the normal Red Hat Linux installation process, RAID arrays can be created. This is done during the disk partitioning phase of the installation. To begin, you must manually partition your disk drives using **Disk Druid**. You will first need to create partitions of the type "software RAID". These partitions will later be combined to form the desired RAID arrays.

Once you have created all the partitions required for the RAID array(s) that you wish to create, you must then use the **RAID** button to actually create the arrays. You will be presented with a dialog box where you select the array's mount point, file system type, RAID device name, RAID level, and the "software RAID" partitions on which this array will be based.

Once the desired arrays have been created, the installation process continues as usual.


Tip

For more information on creating software RAID arrays during the Red Hat Linux installation process, refer to the *Official Red Hat Linux Customization Guide*.

5.5.2.2. After Red Hat Linux Has Been Installed

Creating a RAID array after Red Hat Linux has been installed is a bit more complex. As with the addition of any type of disk storage, the necessary hardware must first be installed and properly configured. Partitioning is a bit different for RAID than it is for single disk drives. Instead of selecting a partition type of "Linux" (type 83) or "Linux swap" (type 82), all partitions that will be part of a RAID array must be set to "Linux raid auto" (type fd). Next, it is necessary to create the `/etc/raidtab` file. This file is responsible for the proper configuration of all RAID arrays on your system. The file format (which is documented in the `raidtab` man page) is relatively straightforward. Here is an example `/etc/raidtab` entry for a RAID 1 array:

```
raiddev             /dev/md0
raid-level                  1
nr-raid-disks               2
chunk-size                  64k
persistent-superblock       1
nr-spare-disks              0
    device          /dev/hda2
    raid-disk     0
    device          /dev/hdc2
    raid-disk     1
```

Some of the more notable sections in this entry are:

- `raiddev` — Shows the special device file name for the RAID array[3]
- `raid-level` — Defines the RAID level to used used by this RAID array
- `nr-raid-disks` — Indicates how many physical disk partitions are to be part of this array
- `nr-spare-disks` — Software RAID under Red Hat Linux allows the definition of one or more spare disk partitions; these partitions can automatically take the place of a malfunctioning disk
- `device`, `raid-disk` — Together, they define the physical disk partitions that will make up the RAID array

Next, it is necessary to actually create the RAID array. This is done with the `mkraid` program. Using our example `/etc/raidtab` file, we would create the `/dev/md0` RAID array with the following command:

```
mkraid /dev/md0
```

3. Note that since the RAID array is composed of partitioned disk space, the device file name of a RAID array does not reflect any partition-level information.

The RAID array /dev/md0 is now ready to be formatted and mounted. This process is no different than the single drive approach outlined in Section 5.4.1.2 and Section 5.4.1.3.

5.5.3. Day to Day Management of RAID Arrays

There is little that needs to be done to keep a RAID array operating. As long as no hardware problems crop up, the array should function just as if it were a single physical disk drive.

However, just as a system administrator should periodically check the status of all disk drives on the system, the RAID arrays should be checked as well.

5.5.3.1. Checking Array Status With /proc/mdstat

The file /proc/mdstat is the easiest way to check on the status of all RAID arrays on a particular system. Here is a sample mdstat (view with the command cat /proc/mdstat):

```
Personalities : [raid1]
read_ahead 1024 sectors
md3 : active raid1 hda4[0] hdc4[1]
      73301184 blocks [2/2] [UU]

md1 : active raid1 hda3[0] hdc3[1]
      522048 blocks [2/2] [UU]

md0 : active raid1 hda2[0] hdc2[1]
      4192896 blocks [2/2] [UU]

md2 : active raid1 hda1[0] hdc1[1]
      128384 blocks [2/2] [UU]

unused devices: <none>
```

On this system, there are four RAID arrays (all RAID 1). Each RAID array has its own section in /proc/mdstat and contains the following information:

- The RAID array device name (minus /dev/)
- The status of the RAID array
- The RAID array's RAID level
- The physical partitions that currently make up the array (followed by the partition's array unit number)
- The size of the array
- The number of configured devices versus the number of operative devices in the array
- The status of each configured device in the array (U meaning the device is OK, and _ indicating that the device has failed)

5.5.3.2. Rebuilding a RAID array with raidhotadd

Should /proc/mdstat show that a problem exists with one of the RAID arrays, the raidhotadd utility program should be used to rebuild the array. Here are the steps that would need to be performed:

1. Determine which disk drive contains the failed partition
2. Correct the problem that caused the failure (most likely by replacing the drive)

3. Partition the new drive so that the partitions on it are *identical* to those on the other drive(s) in the array
4. Issue the following command:
   `raidhotadd <raid-device> <disk-partition>`
5. Monitor `/proc/mdstat` to watch the rebuild take place

Tip

Here is a command that can be used to watch the rebuild as it takes place:

```
watch -n1 cat /proc/mdstat
```

5.6. Monitoring Disk Space

The one system resource that is most commonly over-committed is disk space. There are many reasons for this, ranging from applications not cleaning up after themselves, to software upgrades becoming larger and larger, to users that refuse to delete old email messages.

No matter what the reason, system administrators must monitor disk space usage on an ongoing basis, or face possible system outages and unhappy users. In this section, we will look at some ways of keeping track of disk space.

5.6.1. Using `df`

The easiest way to see how much free disk space is available on a system is to use the `df` command. Here is an example of `df` in action:

```
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/sda3              8428196   4282228   3717836  54% /
/dev/sda1               124427     18815     99188  16% /boot
/dev/sda4              8428196   3801644   4198420  48% /home
none                    644600         0    644600   0% /dev/shm
```

As we can see, `df` lists every mounted file system, and provides information such as device size (under the `1k-blocks` column), as well as the space used and still available. However, the easiest thing to do is to simply scan the `Use%` column for any numbers nearing 100%.

5.7. Implementing Disk Quotas

While it is always good to be aware of disk usage, there are many instances where it is even better to have a bit of control over it. That is what disk quotas can do.

Many times the first thing most people think of when they think about disk quotas is using it to force users to keep their directories clean. While there are sites where this may be the case, it also helps to look at the problem of disk space usage from another perspective. What about applications that, for one reason or another, consume too much disk space? It is not unheard of for applications to fail in ways that cause them to consume all available disk space. In these cases, disk quotas can help limit the damage caused by such errant applications, by forcing it to stop *before* no free space is left on the disk.

5.7.1. Some Background on Disk Quotas

Disk quotas are implemented on a per-file system basis. In other words, it is possible to configure quotas for `/home` (assuming `/home` is on its own file system), while leaving `/tmp` without any quotas at all.

Quotas can be set on two levels:

- For individual users
- For individual groups

This kind of flexibility makes it possible to give each user a small quota to handle "personal" file (such as email, reports, etc.), while allowing the projects they work on to have more sizable quotas (assuming the projects are given their own groups).

In addition, quotas can be set not just to control the number of disk blocks consumed, but also to control the number of inodes. Because inodes are used to contain file-related information, this allows control over the number of files that can be created.

But before we can implement quotas, we should have a better understanding of how they work. The first step in this process is to understand the manner in which disk quotas are applied. There are three major concepts that you should understand prior to implementing disk quotas:

Hard Limit

> The hard limit defines the absolute maximum amount of disk space that a user or group can use. Once this limit is reached, no further disk space can be used.

Soft Limit

> The soft limit defines the maximum amount of disk space that can be used. However, unlike the hard limit, the soft limit can be exceeded for a certain amount of time. That time is known as the *grace period*.

Grace Period

> The grace period is the time during which the soft limit may be exceeded. The grace period can be expressed in seconds, minutes, hours, days, weeks, or months, giving the system administrator a great deal of freedom in determining how much time to give users to get their disk usage below their soft limit.

With these terms in mind, we can now begin to configure a system to use disk quotas.

5.7.2. Enabling Disk Quotas

In order to use disk quotas, you must first enable them. This process involves several steps:

1. Modifying `/etc/fstab`
2. Remounting the file system(s)
3. Running `quotacheck`
4. Assigning quotas

Let us look at these steps in more detail.

5.7.2.1. Modifying `/etc/fstab`

Using the text editor of your choice, simply add the `usrquota` and/or `grpquota` options to the file systems that require quotas:

```
/dev/md0          /                 ext3    defaults        1 1
LABEL=/boot       /boot             ext3    defaults        1 2
none              /dev/pts          devpts  gid=5,mode=620  0 0
LABEL=/home       /home             ext3    defaults,usrquota,grpquota 1 2
none              /proc             proc    defaults        0 0
none              /dev/shm          tmpfs   defaults        0 0
/dev/md1          swap              swap    defaults        0 0
```

In this example, we can see that the `/home` file system has both user and group quotas enabled.

At this point you must remount each file system whose `fstab` entry has been modified. You may be able to simply `umount` and then `mount` the file system(s) by hand, but if the file system is currently in use by any processes, the easiest thing to do is to reboot the system.

5.7.2.2. Running `quotacheck`

When each quota-enabled file system is remounted, the system is now capable of working with disk quotas. However, the file system itself is not yet ready to support quotas. To do this, you must first run `quotacheck`.

The `quotacheck` command examines quota-enabled file systems, building a table of the current disk usage for each one. This table is then used to update the operating system's copy of disk usage. In addition, the file system's disk quota files are updated (or created, if they do not already exist).

In our example, the quota files (named `aquota.group` and `aquota.user`, and residing in `/home/`) do not yet exist, so running `quotacheck` will create them. Use this command:

```
quotacheck -avug
```

The options used in this example direct `quotacheck` to:

- Check all quota-enabled, locally-mounted file systems (`-a`)
- Display status information as the quota check proceeds (`-v`)
- Check user disk quota information (`-u`)
- Check group disk quota information (`-g`)

Once `quotacheck` has finished running, you should see the quota files corresponding to the enabled quotas (user and/or group) in the root directory of each quota-enabled file system (which would be `/home/` in our example):

```
total 44
drwxr-xr-x    6 root     root         4096 Sep 14 20:38 .
drwxr-xr-x   21 root     root         4096 Sep 14 20:10 ..
-rw-------    1 root     root         7168 Sep 14 20:38 aquota.user
-rw-------    1 root     root         7168 Sep 14 20:38 aquota.group
drwx------    4 deb      deb          4096 Aug 17 12:55 deb
drwx------    9 ed       ed           4096 Sep 14 20:35 ed
drwxr-xr-x    2 root     root        16384 Jan 20  2002 lost+found
drwx------    3 matt     matt         4096 Jan 20  2002 matt
```

Now we are ready to begin assigning quotas.

5.7.2.3. Assigning Quotas

The mechanics of assigning disk quotas are relatively simple. The `edquota` program is used to edit a user or group quota:

```
Disk quotas for user ed (uid 500):
  Filesystem       blocks        soft        hard     inodes     soft     hard
  /dev/md3        6618000           0           0      17397        0        0
```

`edquota` uses a text editor (which can be selected by setting the EDITOR environment variable to the full pathname of your preferred editor) to display and change the various settings. Note that any setting left at zero means no limit:

```
Disk quotas for user ed (uid 500):
  Filesystem       blocks        soft        hard     inodes     soft     hard
  /dev/md3        6617996     6900000     7000000      17397        0        0
```

In this example, user ed (who is currently using over 6GB of disk space) has a soft limit of 6.9GB and a hard limit of 7GB. No soft or hard limit on inodes has been set for this user.


Tip

The `edquota` program can also be used to set the per-file system grace period by using the `-t` option.

Although the mechanics of this process are simple, the hardest part of the process always revolves around the limits themselves. What should they be?

A simplistic approach would be to simply divide the disk space by the number of users and/or groups using it. For example, if the system has a 100GB disk drive and 20 users, each user will be given a hard limit of no more than 5GB[4]. That way, each user would be guaranteed 5GB (although the disk would be 100% full at that point).

A variation on this approach would be to institute a soft limit of 5GB, with a hard limit somewhat above that — say 7.5GB. This would have the benefit of allowing users to permanently consume no more than their percentage of the disk, but still permitting some flexibility when a user reaches (and exceeds) their limit.

When using soft limits in this manner, you are actually over-committing the available disk space. The hard limit is 7.5GB. If all 20 users exceeded their soft limit at the same time, and attempted to reach their hard limits, that 100GB disk would actually have to be 150GB in order to allow everyone to reach their hard limit at the same time.

However, in practice not everyone will exceed their soft limit at the same time, making some amount of overcommitment a reasonable approach. Of course, the selection of hard and soft limits is up to the system administrator, as each site and user community is different.

5.7.3. Managing Disk Quotas

There is little actual management required to support disk quotas under Red Hat Linux. Essentially, all that is required is:

4. Although it should be noted that Linux file systems are formatted with a certain percentage (by default, 5%) of disk space reserved for the super-user, making this example less than 100% accurate.

- Generating disk usage reports at regular intervals (and following up with users that seem to be having trouble effectively managing their allocated disk space)
- Making sure that the disk quotas remain accurate

Let us look at these steps in more detail below.

5.7.3.1. Reporting on Disk Quotas

Creating a disk usage report entails running the `repquota` utility program. Using the command `repquota /home` produces this output:

```
*** Report for user quotas on device /dev/md3
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      --   32836       0       0              4     0     0
ed        -- 6617996 6900000 7000000          17397     0     0
deb       --  788068       0       0          11509     0     0
matt      --      44       0       0             11     0     0
```

While the report is easy to read, a few points should be explained. The `--` displayed after each user is a quick way to see whether the block or inode limits have been exceeded. If either soft limit is exceeded, a `+` will appear in place of the `-`; the first character representing the block limit and the second representing the inode limit.

The `grace` columns are normally blank; if a particular soft limit has been exceeded, the column will contain a time specification equal to the amount of time remaining on the grace period. Should the grace period expire, `none` will appear in its place.

Once a report has been generated, the real work begins. This is an area where a system administrator must make use of all the people skills they possess. Quite often discussions over disk space become emotional, as people view quota enforcement as either making their job more difficult (or impossible), that the quotas applied to them are unreasonably small, or that they just do not have the time to clean up their files to get below their quota again.

The best system administrators will take many factors into account in such a situation. Is the quota equitable, and reasonable for the type of work being done by this person? Does the person seem to be using their disk space appropriately? Can you help the person reduce their disk usage in some way (by creating a backup CD-ROM of all emails over one year old, for example)?

Approaching the situation in a sensitive but firm manner is often better than using your authority as system administrator to force a certain outcome.

5.7.3.2. Keeping Quotas Accurate With `quotacheck`

Whenever a file system is not unmounted cleanly (due to a system crash, for example), it is necessary to run `quotacheck`. However, many system administrators recommend running `quotacheck` on a regular basis, even if the system has not crashed.

The command format itself is simple; the options used have been described in Section 5.7.2.2:

```
quotacheck -avug
```

The easiest way to do this is to use `cron`. From the root account, you can either use the `crontab` command to schedule a periodic `quotacheck` or place a script file that will run `quotacheck` in any one of the following directories (using whichever interval best matches your needs):

- `/etc/cron.hourly`

- `/etc/cron.daily`
- `/etc/cron.weekly`
- `/etc/cron.monthly`

Most system administrators choose a weekly interval, though there may be valid reasons to pick a longer or shorter interval, depending on your specific conditions. In any case, it should be noted that the most accurate quota statistics will be obtained by `quotacheck` when the file system(s) it analyzes are not in active use. You should keep this in mind when you schedule your `quotacheck` script.

5.8. A Word About Backups...

One of the most important factors when considering disk storage is that of backups. We have not covered this subject here, because an in-depth section (Section 8.2) has been dedicated to backups.

Getting It Done

redhat.

Chapter 6.

Managing Accounts and Groups

Managing user accounts and groups is an essential part of system administration within an organization. But to manage users effectively, a good system administrator must understand what user accounts and groups are and how they work.

User accounts are used within computer environments to verify the identity of the person using a computer system. By checking the identity of a user, the system is able to determine if the user is permitted to log into the system and, if so, which resources the user is allowed to access.

Groups are logical constructs that can be used to cluster user accounts together for a specific purpose. For instance, if a company has a group of system administrators, they can all be placed in a system administrator group with permission to access key resources and machines. Also, through careful group creation and assignment of privileges, access to restricted resources can be maintained for those who need them and denied to others.

The ability for a user to access a machine is determined by whether or not that user's account exists. Access to an application or file is granted based on the permission settings for the file. The nature of the access users have to their own systems and others on the network should be determined by the organization's system administrators. This helps to ensure the integrity of sensitive information and key resources against accidental or purposeful harm by users.

6.1. User Accounts, Groups, and Permissions

After a normal user account is created, the user can log into the system and access any applications or files they are permitted to access. Red Hat Linux determines whether or not a user or group can access these resources based on the permissions assigned to them.

There are three permissions for files, directories, and applications. The following lists the symbols used to denote each, along with a brief description:

- `r` — Indicates that a given category of user can read a file.
- `w` — Indicates that a given category of user can write to a file.
- `x` — Indicates that a given category of user can execute the file.
- A fourth symbol (–) indicates that no access is permitted.

Each of the three permissions are assigned to three defined categories of users. The categories are:

- *owner* — The owner of the file or application.
- *group* — The group that owns the file or application.
- *everyone* — All users with access to the system.

One can easily view the permissions for a file by invoking a long format listing using the command `ls -l`. For instance, if the user `juan` creates an executable file named `foo`, the output of the command `ls -l foo` would look like this:

```
-rwxrwxr-x    1 juan     juan            0 Sep 26 12:25 foo
```

The permissions for this file are listed are listed at the start of the line, starting with `rwx`. this first set of symbols define owner access. the next set of `rwx` symbols define group access, with the last set of symbols defining access permitted for all other users.

This listing indicates that the file is readable, writable, and executable by the user who owns the file (user `juan`) as well as the group owning the file (which is a group named `juan`). the file is also world-readable and world-executable, but not world-writable.

One important point to keep in mind regarding permissions and user accounts is that every application run on Red Hat Linux runs in the context of a specific user. typically, this means that if user `juan` launches an application, the application runs using user `juan`'s context. however, in some cases the application may need more access in order to accomplish a task. such applications include those that edit system settings or log in users. for this reason, special permissions have been created.

There are three such special permissions within Red Hat Linux. they are as follows:

- *setuid* — used only for applications, this permission indicates that the application runs as the owner of the file and not as the user running the application. it is indicated by the character `s` in place of the `x` in the owner category.
- *setgid* — used only for applications, this permission indicates that the application runs as the group owning the file and not as the group running the application. it is indicated by the character `s` in place of the `x` in the group category.
- *sticky bit* — used primarily on directories, this bit dictates that a file created in the directory can be removed only by the user who created the file. it is indicated by the character `t` in place of the `x` in the everyone category. in Red Hat Linux the sticky bit is set by default on the `/tmp/` directory for exactly this reason.

6.1.1. Usernames and UIDs, Groups and GIDs

Another point worth noting is that user account and group names are primarily for peoples' convenience. Internally, the system uses numeric identifiers. for users, this identifier is known as a *UID*, while for groups the identifier is known as a *GID*. Programs that make user or group information available to users translate the UID/GID values into their more human-readable counterparts. This fact is particularly important when accessing shared media as discussed in Section 6.5.2.1.

Since some system-level programs on Red Hat Linux run under a dedicated UID, and some default system accounts have reserved UID numbers, all UIDs and GIDs below 500 are reserved for system use. For more information on these standard users and groups, see the chapter titled *Users and Groups* in *Official Red Hat Linux Reference Guide*.

When new user accounts are added using a user creation tool such as `/usr/sbin/useradd`, they are assigned the first available UID and GID starting at 500.

User creation tools are discussed further into this chapter. But before reviewing these tools, let us review the files Red Hat Linux uses to define system accounts.

6.2. Files Controlling User Accounts and Groups

On Red Hat Linux, information about user accounts and groups are stored in several text files within the `/etc/` directory. When a system administrator creates new user accounts, these files must either be edited by hand or applications must be used to make the necessary changes.

The following section document the files in the `/etc/` directory that store user and group information under Red Hat Linux.

6.2.1. `/etc/passwd`

The `/etc/passwd` file is world-readable, and contains a list of users, each on a separate line. On each line is a seven field, colon delimited list which contains the following information:

- *Username* — The name the user types when logging into the system.
- *Password* — This contains the encrypted password for the user (or an `x` if shadow passwords are being used — more on this later).
- *User ID* (*UID*) — The numerical equivalent of the username which is referenced by the system and applications when determining access privileges.
- *Group ID* (*GID*) — The numerical equivalent of the primary group name which is referenced by the system and applications when determining access privileges.
- *GECOS* — The GECOS[1] field is optional, and is used to store extra information (such as the user's full name). Multiple entries can be stored here in a comma delimited list. Utilities such as `finger` access this field to provide additional user information.
- *Home directory* — The absolute path to the user's home directory, such as `/home/juan`.
- *Shell* — The program automatically launched whenever a user logs in. This is usually a command interpreter (often called a *shell*). Under Red Hat Linux, the default value is `/bin/bash`. If this field is left blank, `bin/sh` is used. If it is set to a non-existent file, then the user will be unable to log into the system.

Here is an example of a `/etc/passwd` entry:

```
root:x:0:0:root:/root:/bin/bash
```

This line shows that the `root` user has a shadow password, as well as a UID and GID of 0. The `root` user has `/root/` as a home directory, and uses `/bin/bash` for a shell.

For more information about `/etc/passwd`, type `man 5 passwd`.

6.2.2. `/etc/shadow`

The `/etc/shadow` file is readable only by the root user, and contains password and optional password aging information. As in the `/etc/passwd` file, each user's information is on a separate line. Each of these lines is a nine field, colon delimited list including the following information:

- *Username* — The name the user types when logging into the system. This allows the **login** application to retrieve the user's password (and related information).
- *Encrypted password* — The 13 to 24 character password. The password is encrypted using either the `crypt` library function, or the md5 hash algorithm. In this field, values other than a validly-formatted encrypted or hashed password are used to control user logins and to show the password status. For example, if the value is `!` or `*` the account is locked, and the user is not allowed to log in. If the value is `!!` a password has never been set before (and the user, not having set a password, will not be able to log in).
- *Date password last changed* — The number of days since January 1, 1970 (also called the *epoch*) that the password was last changed. This information is used for the following password aging fields.
- *Number of days before password can be changed* — The minimum number of days that must pass before the password can be changed.
- *Number of days before password change is required* — The number of days that must pass before the password must be changed.
- *Number of days warning before password change* — The number of days before password expiration during which the user is warned of the impending expiration.

1. GECOS stands for General Electric Comprehensive Operating System.

- *Number of days before the account is disabled* — The number of days after a password expires before the account will be disabled.
- *Date since the account has been disabled* — The date (stored as the number of days since the epoch) since the user account has been disabled.
- *A reserved field* — A field that is ignored in Red Hat Linux.

Here is an example line from `/etc/shadow`:

```
juan:$1$.QKDPc5E$SWlkjRWexrXYgc98F.:11956:0:90:5:30:12197:
```

This line shows the following information for user `juan`:

- The password was last changed September 25, 2002
- There is no minimum amount of time required before the password can be changed
- The password must be changed every 90 days
- The user will get a warning five days before the password must be changed.
- The account will be disabled 30 days after the password expires if no login attempt is made
- The account will expire on May 24, 2003

For more information on the `/etc/shadow` file, type `man 5 shadow`.

6.2.3. `/etc/group`

The `/etc/group` is world-readable, and contains a list of groups, each on a separate line. Each line is a four field, colon delimited list including the following information:

- *Group name* — The name of the group. Used by various utility programs to identify the group.
- *Group password* — If set, this allows users who are not part of the group to join the group by using the `newgrp` command and typing the password stored here. If a lower case `x` is in this field, then shadow group passwords are being used.
- *Group ID* (*GID*) — The numerical equivalent of the group name. It is used by the system and applications when determining access privileges.
- *Member list* — A comma delimited list of users in the group.

Here is an example line from `/etc/group`:

```
general:x:502:juan,shelley,bob
```

This line shows that the general group is using shadow passwords, has a GID of 502, and that `juan`, `shelley`, and `bob` are members.

For more information on `/etc/group`, type `man 5 group`.

6.2.4. `/etc/gshadow`

The `/etc/gshadow` file is readable only by the root user, and contains an encrypted password for each group, as well as group membership and administrator information. Just as in the `/etc/group` file, each group's information is on a separate line. Each of these lines is a four field, colon delimited list including the following information:

- *Group name* — The name of the group. Used by various utility programs to identify the group.

- *Encrypted password* — The encrypted password for the group. If set, non-members of the group can join the group by typing the password for that group using the `newgrp` command. If the value is of this field `!` then no user is allowed to access the group using the `newgrp` command. A value of `!!` is treated the same as a value of `!` only it indicates that a password has never been set before. If the value is null, only group members can log into the group.
- *Group administrators* — Group members listed here (in a comma delimited list) can add or remove group members using the `gpasswd` command.
- *Group members* — Group members listed here (in a comma delimited list) are regular, non-administrative members of the group.

Here is an example line from `/etc/gshadow`:

```
general:!!:shelley:juan,bob
```

This line shows that the `general` group has no password and does not allow non-members to join using the `newgrp` command. In addition, `shelley` is a group administrator, and `juan` and `bob` are regular, non-administrative members.

Since editing these files by hand raises the potential for syntax errors, it is recommended that the applications provided with Red Hat Linux for this purpose be used instead. The next section reviews the primary tools for performing these tasks.

6.3. User Account and Group Applications

There are two basic types of applications one can use when managing user accounts and groups on Red Hat Linux systems:

- The graphical **User Manager** application
- A suite of command line tools

For detailed instructions on using **User Manager**, see the chapter titled *User and Group Configuration* in the *Official Red Hat Linux Customization Guide*.

While both the **User Manager** application and the command line utilities perform essentially the same task, the command line tools have the advantage of being scriptable and therefore, more easily automated.

The following table describes some of the more common command line tools used to create and manage users:

Application	Function
`/usr/sbin/useradd`	Adds user accounts. This tool is also used to specify primary and secondary group membership.
`/usr/sbin/userdel`	Deletes user accounts.
`/usr/sbin/usermod`	Edits account attributes including some functions related to password aging. For more fine-grained control, use the `passwd` command. `usermod` is also used to specify primary and secondary group membership.
`passwd`	Sets passwords. Although primarily used to change a user's password, it also controls all aspects of password aging.

Application	Function
`/usr/sbin/chpasswd`	Reads in a file consisting of username and password pairs, and updates each users' password accordingly.
`chage`	Changes the user's password aging policies. The `passwd` command can also be used for this purpose.
`chfn`	Changes the user's GECOS information.
`chsh`	Changes the user's default shell.

Table 6-1. User Management Command Line Tools

The following table describes some of the more common command line tools used to create and manage groups:

Application	Function
`/usr/sbin/groupadd`	Adds groups, but does not assign users to those groups. The `useradd` and `usermod` programs should then be used to assign users to a given group.
`/usr/sbin/groupdel`	Deletes groups.
`/usr/sbin/groupmod`	Modifies group names or GIDs, but does not change group membership. The `useradd` and `usermod` programs should be used to assign users to a given group.
`gpasswd`	Changes group membership and sets passwords to allow non-group members who know the group password to join the group. It is also used to specify group administrators.
`/usr/sbin/grpck`	Checks the integrity of the `/etc/group` and `/etc/gshadow` files.

Table 6-2. Group Management Command Line Tools

The tools listed thus far provide system administrators great flexibility in controlling all aspects of user accounts and group membership. To learn more about how they work, refer to the man page for each. These applications do not, however, determine what resources these users and groups have control over. For this, the system administrator must use file permission applications.

6.3.1. File Permission Applications

Permissions for files, directories, and applications are an integral part of managing resources within an organization. The following table describes some of the more common command line tools used for this purpose.

Application	Function
`chgrp`	Changes which group owns a given file.
`chmod`	Changes access permissions for a given file. It is also capable of assigning special permissions.
`chown`	Changes a file's ownership (and can also change group).

Table 6-3. Permission Management Command Line Tools

It is also possible to alter these attributes in GNOME and KDE graphical environments by right-

clicking on the desired object and selecting **Properties**. The next section will review what happens when an application is used to create user accounts and groups.

6.4. The Process of Creating User Accounts

When you create a user account using the **User Manager** application, you can manage all aspects of the user account. For detailed instructions on using **User Manager**, see the chapter titled *User and Group Configuration* in the *Official Red Hat Linux Customization Guide*. This section will highlight the multi-step user creation process necessary when using the command line tools.

There are two steps to creating a user with the command line tools included with Red Hat Linux:

1. Issue the `useradd` command to create a locked user account.
2. Unlock the account by issuing the `passwd` command to assign a password and set password aging guidelines.

The following steps illustrate what happens if the command `/usr/sbin/useradd juan` is issued on a system that has shadow passwords enabled:

1. A new line for `juan` is created in `/etc/passwd`. The line has the following characteristics:
   - It begins with the username, `juan`.
   - There is an `x` for the password field indicating that the system is using shadow passwords.
   - A UID at or above 500 is created. (Under Red Hat Linux UIDs and GIDs below 500 are reserved for system use.)
   - A GID at or above 500 is created.
   - The optional GECOS information is left blank.
   - The home directory (`/home/juan/`) is specified.
   - The default shell is set to `/bin/bash`.

2. A new line for a group named `juan` is created in `/etc/shadow`. The line has the following characteristics:
   - It begins with the username, `juan`.
   - Two exclamation points (`!!`) appear in the password field of the `/etc/shadow` file, which locks the account.
   - The password is set to never expire.

3. A new line for a group named `juan` is created in `/etc/group`. A group bearing the same name as a user is called a *user private group*. For more information on user private groups, see the chapter titled *Users and Groups* in the *Official Red Hat Linux Reference Guide*.

   The line created in `/etc/group` has the following characteristics:
   - It begins with the group name, `juan`.
   - An `x` appears in the password field indicating that the system is using shadow group passwords.
   - The GID matches the one listed for user `juan` in `/etc/passwd`.

4. A new line for a group named `juan` is created in `/etc/gshadow`. The line has the following characteristics:
   - It begins with the group name, `juan`.
   - Two exclamation points (`!!`) appear in the password field of the `/etc/gshadow` file, which locks the group.
   - All other fields are blank.
5. A directory for user `juan` is created in the `/home/` directory. This directory is owned by user `juan` and group `juan`. However, it has read, write, and execute privileges *only* for the user `juan`. All other permissions are denied.
6. The files within the `/etc/skel/` directory (which contain default user settings) are copied into the new `/home/juan/` directory.

At this point, a locked account called `juan` exists on the system. To activate it, the administrator must next assign a password to the account using the `passwd` command and, optionally, set password aging guidelines.

It is also possible to configure the account so that during the first log in, the user is asked to create a password. See Section 6.4.2.

6.4.1. Password Security

Creating strong passwords is important for the security of the organization. There are two options available to enforce the use of good passwords:

- The system administrator can create passwords for all users.
- The system administrator can let the users create their own passwords, while verifying that the passwords are of acceptable quality.

Creating passwords for the users ensures that the passwords are good, but it becomes a daunting task as the organization grows.

It also increases the risk of users writing their passwords down.

For these reasons, system administrators prefer to have the user create their own passwords. However, a good system administrator actively verifies that the passwords are good and, in some cases, forces users to change their passwords periodically through password aging.

For guidelines on how to create strong passwords and how to set password aging policies, see the chapter titled *Workstation Security* in the *Official Red Hat Linux Security Guide*.

6.4.2. New User Passwords

If passwords within an organization are created centrally by the administrator, adding new users to the organization means the administrators must configure the account so the user is asked to create a password when logging in for the first time.

To configure a user account in this manner, follow these steps:

1. *Create the user account using the `useradd` command.* — At this point the account is created, but locked.
2. *Force immediate password expiration* — To do this. type the following command:
   ```
   chage -d 0
   ```

This sets the value for the date the password was last changed to the epoch (January 1, 1970). This value forces immediate password expiration no matter what password aging policy, if any, is in place.

3. *Unlock the account* — There are two common approaches to this. The administrator can assign an initial password:
```
/usr/sbin/usermod -p "<password>"
```
In the above command, replace <password> with the initial password.

Or, the administrator can assign a null password:
```
/usr/sbin/usermod -p ""
```

Caution

While using a null password is convenient for both the user and the administrator, there is a slight risk that a third party can log in first and access the system. To minimize this threat, it is recommended that administrators verify that user is ready to log in when they unlock the account.

In either case, upon initial log in, the user is prompted for a new password.

6.5. Managing User Resources

Creating user accounts is only part of a system administrator's job. Management of user resources is also essential. Therefore, three points must be considered:

- Who can access shared data.
- Where users access this data.
- What barriers are in place to prevent abuse of resources.

This section will briefly review each of these topics.

6.5.1. Who Can Access Shared Data

The identity of those who can access a given application, file, or directory on a system is determined by its permissions. By default, Red Hat Linux places reasonable permissions on the file system.

For instance, the `/tmp` directory, which is world-writable, also has the sticky bit set. This means that only the user who writes a file to the directory can delete it. This prevents other users from mistakenly or maliciously deleting the files of others.

Another example are the permissions assigned by default to a user's home directory. Only the owner of the home directory can create or view files there. Other users on the system are denied all access. This increases user privacy and prevents possible misappropriation of personal files.

But there are many situations where multiple users may need access to the same resources on a machine. In this case, careful creation of shared groups may be necessary.

6.5.1.1. Groups and Shared Data

As mentioned in the introduction, groups are logical constructs that can be used to cluster user accounts together for a specific purpose. When managing users within an organization, it is wise to identify what data should be accessed by certain departments, what data should be denied to others, and what data should be shared by all. Determining this will aid in creating an appropriate group structure, along with permissions appropriate for the shared data.

For instance, let us say that the Accounts Receivable department needs to maintain a list of accounts that are delinquent on their payments. They must also share that list with the Collections department. If both Accounts Receivable and Collections personnel are placed in a group called `accounts`, this information can then be placed in a shared directory (owned by group `accounts`) with group read and write permissions on the directory.

6.5.1.2. Determining Group Structure

Some of the challenges facing system administrators when creating shared groups are:

- What groups to create?
- Who to put in a given group?
- What type of permissions should these shared resources have?

A common sense approach to these questions is helpful. For instance, it is often best to mirror the organizational structure when creating groups. For instance, if there is a Finance department, then create a group called `finance`, and make all Finance personnel member of that group. If the financial information is too sensitive for the company at large, but vital for senior officials within the organization, then grant the senior officials group permission to access the directories and data used by the finance department by adding all senior officials to the `finance` group.

It is also good to err on the side of caution when granting permissions to users. This way, sensitive information is less likely to fall into the wrong hands.

By approaching the group structure for an organization in this way, the need for access to share data within the organization can be safely and effectively met.

6.5.2. Where Users Access Shared Data

When sharing data among users, it is common practice to have a central server (or group of servers) that export certain directories to other machines on the network. This way data is stored in one place; synchronizing data between multiple machines is not necessary.

One of the best ways to share directories under Red Hat Linux is to use NFS. For more information on mounting and exporting directories using NFS, see the chapter titled *Network File System (NFS)* in the *Official Red Hat Linux Customization Guide*.

If your network includes other platforms, you may also find it necessary to share directories using Samba. See the chapter titled *Samba* in the *Official Red Hat Linux Customization Guide* for more information.

Unfortunately, once data is shared between multiple computers on a network, the potential for conflicts in file ownership can arise.

6.5.2.1. The UID/GID Conundrum

As mentioned in Section 6.1.1, user account and group names are primarily for peoples' convenience. All other applications and processes on the system only operate on UIDs and GIDs. This fact is normally transparent to users, until a user accesses a shared volume. If the `/etc/passwd` and `/etc/group` files on the file server and the user's machine differ in the UIDs or GIDs they contain, improper application of permissions can lead to security issues.

For example, if user `juan` has a UID of 500 on a desktop computer, files `juan` creates on a file server will be created with owner UID 500. However, if user `bob` logs in locally to the file server, and `bob`'s account also has a UID of 500, `bob` will have full access to `juan`'s files, and vice versa.

One of the best ways to avoid this issue is to centrally manage the `/etc/passwd` and `/etc/group` files using *Network Information Services* (*NIS*), *Lightweight Directory Access Protocol* (*LDAP*), or *Hesiod*. This way all users on the network share the same user database and ownership conflicts on shared volumes are eliminated.

Caution

Although it is a good idea to centralize the information contained in `/etc/passwd` using these mechanisms, for security reasons it is not a good idea to use them to centralize the information contained in the `/etc/shadow` or the `/etc/gshadow` files. Instead, consider using Kerberos for this purpose. See the chapter titled *Kerberos* in the *Official Red Hat Linux Customization Guide* for more information on how to accomplish this.

6.5.2.2. Home Directories

Another issue facing administrators is whether or not users should have centralized home directories.

By default, Red Hat Linux creates home directories for new users in the `/home/` directory. This is fine for stand alone machines, but within an organization, it may be helpful to users to have home directories centralized.

The primary advantage of centralizing home directories on a network-attached server is that if a user logs into any machine on the network, they will be able to access the files in their home directory. The disadvantage is that if the network goes down, users across the entire organization will be unable to get to their files.

In some situations (such as laptops), having centralized home directories may not be practical. But if the administrator deems it appropriate, either NIS or LDAP (in conjunction with NFS) is an effective means of implementing centralized home directories.

6.5.3. What Barriers Are in Place To Prevent Abuse of Resources

Careful organization of groups and assignment of permissions for shared resources is one of the most important things an administrator can do to prevent abuse among users within an organization. This way those who should not have access to sensitive resources are denied access.

But the best guard against abuse of resources is always sustained vigilance on the part of the systems administrator.

Chapter 7.

Printers and Printing

Printers are an essential resource for creating a *hard copy* — a physical reproduction using paper — of documents and collateral for business, academic, and home users. It has become an indispensable peripheral in all levels of business and institutional computing. This chapter will discuss the various printers available and compare their uses in different computing environments. It will then discuss the configuration of Red Hat Linux to work with local and networked printers.

7.1. Types of Printers

Like any other computer peripheral, there are several types of printers available for your use. Some printers employ technologies that mimic manual typewriter-style functionality, while others use sprayed organic ink or electrically-charged powder medium. Printer hardware interfaces with a PC or network using parallel, serial, or data networking protocols. There are several factors to consider when evaluating printers for procurement and deployment in your computing environment.

The following sections discuss the various printer types and the protocols that printers use to communicate with computers.

7.1.1. Printing Considerations

There are several aspects to factor into printer evaluations. The following specifies some of the most common criteria when evaluating your printing needs.

7.1.1.1. Function

Evaluating your organizational needs and how a printer services those needs is the essential criteria in determining the right type of printer for your environment. The most important question to ask is "*What do we need?*". Since there are specialized printers for either text, images, or any variation thereof, you should be certain that you procure the right tool for your purposes.

For example, if your requirements call for high-quality color images on professional-grade glossy paper, it is recommended to use a dye-sublimation or thermal wax transfer color printer than a laser or impact printer.

Conversely, a laser or inkjet printers are well-suited for printing rough drafts or documents intended for internal distribution (such high-volume printers are usually called *workgroup* printers). Determining the needs of the everyday user allows administrators to determine the right printer for the job.

Other factors to consider are features such as *duplexing* — the ability of the printer to print on both sides of a piece of paper. Traditionally, printers could only print on one side of the page (called *simplex* printing). Most printer models today do not have this feature by default (or it may be able to simulate a manual duplexing method which forces the user to flip the paper themselves). Some models offer add-on hardware for duplexing; however such add-ons can drive one-time costs up considerably. However duplex printing may reduce costs over time by reducing the amount of paper used to print documents, thus reducing recurring consumables costs.

Another factor to consider is paper size. Most printers are capable of handling *letter* (8 1/2" x 11") and *legal* (8 1/2" x 14") sized paper. If certain departments (such as marketing or design) have specialized needs such as creating posters or banners, there are *large-format* printers which can handle tabloid (11" x 17") sizes or larger.

Additionally, high-end features such as network modules for workgroup and remote site printing should also be considered during evaluation. More information about networked printing can be found in Section 7.7.

7.1.1.2. Cost

Cost is another factor to consider when evaluating printers. However, simply determining the one-time cost associated with the purchase of the printer itself is not enough of a determinant. There are other costs to consider, such as *consumables*, parts and maintenance, and printer add-ons.

Consumables is a general term for printing supplies. *Ink* and *paper* are the two most common printing consumables. Ink is the material that the printer projects onto the *medium* (the paper).

Ink is, itself, a generalized term, as not all printers function using standard, water-based inks. Laser printers use powder, while impact printers use ribbons. There are specialized printers that heat the ink before it is transferred onto paper, while others spray the ink in small drops onto the printing surface. Ink replacement costs vary widely and depend on whether the ink can be *recharged* (refilled) by hand or if it requires a full replacement of the ink *cartridge* (the ink housing).

There are also various types of paper or print medium to choose from. For most printing needs, a wood-pulp based paper medium is sufficient. However, there are variations of paper that are recommended (or required) for certain printers. For example, creating accurate prints of digital images require a special glossy paper suitable for high exposure to natural or artificial lighting as well as accuracy; such qualities are known as color *fastness*. For *archival-quality* documents that require durability and a professional level of legibility (such as contracts, résumés, and permanent records), a *matted* (or non-glossy) paper medium should be used. The *stock* (or thickness) of paper is also important, as some printers do not feed paper completely straight during the printing process, which can cause jams on plain paper printers. Some printers can also print on *transparencies* — a thin film that allows light from a projector to pass through it and display the resulting imprinted image on to a projected surface for presentations and lectures. Specialized papers such as those noted here can affect consumables costs, and should be taken in consideration when evaluating printing needs.

7.2. Impact Printers

Impact printers are the oldest print technologies still in active production. Some of the largest printer vendors continue to manufacture, market, and support impact printers, parts, and supplies. Impact printers are most functional in specialized environments where low-cost printing is essential. The two most common forms of impact printers are *dot-matrix* and *daisy-wheel*.

7.2.1. Dot-Matrix Printers

The technology behind dot-matrix printing is quite simple. The paper is pressed against a *drum* (a rubber-coated cylinder) and is intermittently pulled forward as printing progresses. The electromagnetically-driven *print head* moves across the paper and strikes the printer ribbon situated between the paper and printhead pin. The impact of the printhead against the printer ribbon imprints ink dots on the paper which form human-readable characters.

Dot-matrix printers vary in print resolution and overall quality with either 9 or 24-pin printheads. The more pins per inch, the higher the print resolution. Most dot-matrix printers had a maximum resolution around 240 *dpi* (dots per inch). While this resolution is not as high as those possible in laser or Inkjet printers, there is one distinct advantage to dot-matrix (or any form of impact) printing. Because the the printhead strikes the surface of the paper with enough force to embed characters on the page, it is ideal for environments that frequently print on *carbon copy*, special multi-sheeted documents with carbon on the underside that will create a mark on the sheet underneath it when enough pressure is applied. Retailers and small businesses often use carbon copy as receipts or bills of sale.

7.2.2. Daisy-wheel Printers

If you have ever seen or worked with a manual typewriter before, then you understand the technological concept behind daisy-wheel printers. These types of printers have printheads composed of multi-plated metallic or plastic wheels cut into *petals*. Each petal has a letter (in capital and lowercase), number, or punctuation mark on it that is raised. When the petal is struck against the printer ribbon, the resulting letter is embedded in ink onto the paper. Daisy-wheel printers are loud and slow. They cannot print graphics, cannot change fonts unless the wheel is physically replaced with a wheel of a different font, and are generally not used in modern computing environments. However, Red Hat Linux does include the Common UNIX Printing System (CUPS), which has a comprehensive printer compatibility list in case your environment requires use of daisy-wheel printers.

7.2.3. Line Printers

Another type of impact printer somewhat related to daisy-wheel is the *line printer*, which has multiple columns of characters lined up instead of a petaled wheel. As the roller moves the paper forward one line, the appropriate characters strike a ribbon onto the paper, causing an entire line to be printed at one time, rather than one character or area of text. Line printers are much faster than dot-matrix or daisy-wheel printing; however, they are quite loud and produce lower print quality.

7.2.4. Impact Printer Consumables

Ink ribbons and paper are the primary recurring costs of impact printers. Of all the printer types, however, impact printers have relatively low consumable costs. Impact printers require a continuous, uncut ream of paper that has perforations between each page. Pre-punched holes on either side of the page help the paper move against the print drum smoothly, preventing paper jams or print misalignment.

7.3. Inkjet Printers

inkjet is one of the most popular printing technologies today. The relative low cost of the printers and multi-purpose printing abilities make it a good choice for small businesses and home offices.

Inkjets use quick-drying, water-based inks and a printhead with a series of small nozzles that spray ink on the surface of the paper. The printhead assembly is driven by a belt-fed motor that moves the print head across the paper.

Inkjets were originally manufactured to print in *monochrome* (black and white) only. However, the printhead has since been expanded and the nozzles increased to accommodate cyan, magenta, and yellow. This combination of colors (called *CMYK*) allows for printing images with nearly the same quality as a photo development lab using certain types of coated paper. When coupled with crisp and highly readable text print quality, inkjet printers are a sound all-in-one choice for monochrome or color printing needs.

7.3.1. Inkjet Consumables

Inkjet printers tend to be low-cost and scale slightly upward based on print quality, extra features, and abilities to print on larger formats than the standard legal or letter paper sizes. While the one-time cost of purchasing an inkjet is lower than other printer types, there is the factor of inkjet consumables that must be considered. Because demand for inkjets is large and spans the computing spectrum from home to enterprise, the procurement of consumables can be costly. Note that with most CMYK inkjet printers, separate ink cartridges for each color must be purchased, although some use one cartridge for CMY and another cartridge for K (black).

Some inkjet manufacturers also require you to use specially treated paper for printing high-quality images and documents. Such paper uses a moderate to high gloss coating formulated to absorb colored inks, which prevents *clumping* (the tendency for water-based inks to collect in certain areas where colors blend, causing muddiness or dried ink blots) or *banding* (where the print output has *striped* pattern's of extraneous lines on the printed page. Consult the printer manufacturer manual about recommended papers.

7.4. Laser Printers

An older technology than inkjet, laser printers are another popular alternative to legacy impact printing. Laser printers are known for their high volume output and low cost-per-page. Laser printers are often deployed in enterprises as a workgroup or departmental print center, where performance, durability, and output requirements are a constant. Because laser printers service these needs so readily (and at a reasonable cost-per-page), the technology is widely regarded as the workhorse of enterprise printing.

Laser printers share much of the same (or similar) technologies as photocopiers. Mechanized rollers and gears pull a sheet of paper from a paper tray and through a *charge roller*, which infuses the paper with an electrostatic charge. The paper then passes through a printing drum, which is itself inversely charged and scanned by a laser that emits the print contents across the drum, discharging the drum at points corresponding to text and image points. The laser receives the print information from the laser printer's microprocessor, which in some cases can be a powerful *RISC* (Reduced Instruction Set Computer) architecture processor used for complex image rendering of print jobs. The processor *rasterizes* (or creates bitmap images from vector primitives and/or typographical/layout expressions) the print job and directs the laser to recreate a reproduction onto the drum. Then, as the paper passes through the drum, the charge of the paper reacts with the inversely charged drum. *Toner* (special powdered ink) is sprinkled on the drum and is pulled off of the drum as the paper passes through. Finally, the paper is passed through *fusing rollers*, which heat the paper and melts the toner (which would otherwise slide off the page as it exits the printer) onto the paper. The paper exits the printer literally *hot off the press*.

7.4.1. Color Laser Printers

Color laser printers are an emerging technology created by printer manufacturers whose aim is to collect the features of laser and inkjet technology into a multi-purpose printer package. The technology is based on traditional monochrome laser printing, but uses additional technologies to create color images and documents. Instead of using black toner only, color laser printers use a CMYK toner combination. The print drum either rotates each color and lays the toner down one color at a time, or lays all four colors down onto a plate and then passes the paper through the drum, transferring the complete image onto the paper. Color laser printers also employ *fuser oil* along with the heated fusing rolls, which further bonds the color toner to the paper and can give varying degrees of gloss to the image finish.

Because of its increased features, color laser printers are typically twice (or several times) as expensive as a monochrome laser printer. In calculating the total cost of ownership with regards to printing resources, some administrators may wish to separate monochrome (text) and color (image) functionality to a dedicated monochrome laser printer and a dedicated inkjet printer, respectively.

7.4.2. Laser Consumables

Depending on the type of laser printer deployed, consumable costs usually are fixed and scale evenly with increased usage or print job volume over time. Toner comes in cartridges that are usually replaced outright; however, some models come with refillable cartridges. Color laser printers require one toner cartridge for each of the four colors. Additionally, color laser printers require fuser oils to bond toner

onto paper and waste toner bottles to capture toner spillover. These added supplies raise the consumables cost of color laser printers; however, it is worth noting that such consumables, on average, last about 6000 pages, which is much greater than comparable inkjet or impact consumable lifespans. Paper type is less of an issue in laser printers, which means bulk purchases of regular xerographic or photocopy paper is acceptable for most print jobs; however, if you plan to print high-quality images, you should opt for glossy paper for a professional finish.

7.5. Other Printer Types

There are other types of printers available, mostly special-purpose printers for professional graphics or publishing organizations. These types of printers are not for general purpose use, however. Because they are relegated to niche uses, their prices (one-time and recurring consumables costs) are higher, as relative to impact, laser, and inkjet printers.

Thermal Wax Printers

Used mostly for business presentation transparencies and for color *proofing* (creating test documents and images for close quality inspection before sending off master documents to be pressed on industrial four-color offset printers). Thermal wax printers use sheet-sized, belt driven CMYK ribbons and specially-coated paper or transparencies. The printhead contains heating contacts that melt each colored wax onto the paper as it is rolled through the printer.

Dye-Sublimation Printers

Used in organizations such as service bureaus — where professional quality documents, pamphlets, and presentations are more important than consumables costs — dye-sublimation (or dye-sub) printers are the workhorses of quality CMYK printing. The concepts behind dye-sub printers are similar to thermal wax printers except for the use of diffusive plastic dye film instead of colored wax as the *ink* element. The printhead heats the colored film and vaporizes the image onto specially coated paper.

Dye-sub is quite popular in the design and publishing world as well as the scientific research field, where preciseness and detail are required. Such detail and print quality comes at a price, as dye-sub printers are also known for their high costs-per-page.

Solid Ink Printers

Used mostly in the packaging and industrial design industries, solid ink printers are prized for their ability to print on several paper stocks. Solid ink printers, as the name implies, use hardened ink sticks that that are melted and sprayed through small nozzles on the printhead. The ink is then sent through a fuser roller which further absorbs the ink onto the paper.

The solid ink printer is ideal for prototyping and proofing new designs for product packages; as such, most service-oriented businesses would not find a need for such a niche printer type.

7.6. Printer Languages and Technologies

Before the advent of laser and inkjet technology, impact printers could only print standard, justified text with no variation in letter size or font style. Today, printers are able to process complex documents with embedded images, charts, and tables in multiple frames and in several languages, all in one print job. Such complexity must adhere to some format conventions. This is what spurred the development of the *page description language* (or PDL) — a specialized document formatting language specially made for computer communication with printers.

Over the years, printer manufacturers have developed their own proprietary languages to describe document formats. However, such proprietary languages are applied only to the printers that the man-

ufacturers created themselves. If, for example, you were to send a print-ready file using a proprietary PDL to a professional press, there was no guarantee that your file would be compatible with the printer's machines. The issue of portability came into question.

Xerox® developed the Interpress™ protocol for their line of printers, but full adoption of the language by the rest of the printing industry was never realized. Two original developers of Interpress left Xerox and formed Adobe®, a software company catering mostly to electronic graphics and document professionals. At Adobe, they developed a widely-adopted PDL called *PostScript*™, which uses a markup language to describe text formatting and image information that could be processed by printers. At the same time, the Hewlett-Packard® Company developed the *Printer Control Language*™ (or PCL) for use in their ubiquitous Laser and inkjet printer lines. Postscript and PCL are widely adopted PDLs and are supported by most printer manufacturers (along with the printer's own proprietary languages, when available).

PDLs work on the same principle as computer programming languages. When a document is ready for printing, the PC or workstation takes the images, typographical information, and document layout, and uses them as objects that form instructions for the printer to process. The printer then translates those objects into *rasters* — a series of scanned lines that form an image of the document (called *Raster Image Processing* or RIP), and prints the output onto the page as one image, complete with text and any graphics included. This work-flow makes printing documents of any complexity uniform and standard, allowing for little or no variation in printing from one printer to the next. PDLs are designed to be portable to any format, and scalable to fit several paper sizes.

Choosing the right printer is a matter of determining what standards the various departments in your organization have adopted for their needs. Most departments use word processing and other productivity software that use the postscript language for outputting to printers. However, if your graphics department require PCL or some proprietary form of printing, you must take that into consideration, as well.

7.7. Networked Versus Local Printers

Depending on organizational needs, it may be unnecessary to assign one printer to each member of your organization. Such overlap in expenditure can eat into allotted budgets, leaving less capital for other necessities. While local printers attached via parallel or USB cable to every workstation are an ideal solution, it is not often feasible economically.

Printer manufacturers have addressed this need by developing *departmental* (or workgroup) printers. These machines are usually durable, fast, and have long-life consumables. Workgroup printers usually are attached to a print server, a standalone device (such as a reconfigured workstation) which handles print jobs and routes output to the proper printer when available (although, some printers include built-in or add-on network interfaces that eliminate the need for a dedicated print server). Print servers can use either the *Internet Printing Protocol* (IPP) available in Red Hat Linux via the *Common UNIX Printing System* (CUPS) or through Samba. Samba is particularly useful for heterogeneous environments where departments may use different operating systems. More information about configuring Red Hat Linux for use as a print server can be found in Section 7.9.

7.8. Printer Configuration and Setup

Once you have evaluated your needs and have procured your printers and supplies, it is now time to deploy them in your computing environment. Red Hat Linux has many utilities designed to ease the deployment cycle and get your organization printing quickly. Red Hat Linux includes several configuration tools for printing and networking that you can use in a graphical or command line environment. Additionally, the configuration tools included can be administered remotely for added convenience if issues arise while you are away.

The **Printer Configuration Tool** is a utility for Red Hat Linux that allows administrators to add, edit, and configure printer hardware to work in a local or networked environment. The interface is designed

to make printer setup and administration less complex than editing configuration files manually. Administrators can use the **Printer Configuration Tool** on local printers attached to workstations, and even share such printers to other users on the network. The **Printer Configuration Tool** uses an IPP-compatible backend well-suited for network printing. Figure 7-1 shows the **Printer Configuration Tool** in action.

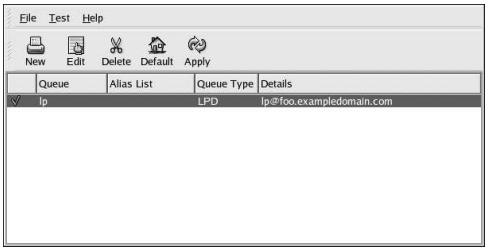

Figure 7-1. The Printer Configuration Tool

7.8.1. An Example Setup

If you are administrating a local network and wish to allow users to print to a laser printer, the most ideal solution would be a laser printer with an Ethernet port to hook up to your network. If you have multiple printers (or a printer without network access in the hardware), you can create a print server that handles all print jobs and queues the jobs based on arbitrary criteria. A decommissioned workstation or server is a perfectly suitable machine to be re-purposed for print serving. Plug your printer (by parallel port or USB, if supported) into your server, install Red Hat Linux, and choose an installation type that will install the proper printer services on your system. If you desire a more streamline selection of packages for the print server, choose a **`Custom`** installation type. If you are not sure what packages you will need, choose the **`Server`** installation type, which will install all the packages needed to configure a printer and setup a connection to share the printer to clients on the network via Samba or the printer daemon. For more information about installing Red Hat Linux on your system, refer to the *Official Red Hat Linux Installation Guide*.

Once you have your system running Red Hat Linux, you can now begin printer configuration. You may use either the text or graphical version of the **Printer Configuration Tool** (you must have the X Window System [or X] installed and loaded in order to use the graphical version). Start the tool at a shell prompt by typing `redhat-config-printer-tui` for the text version. For the graphical version, if you have the graphical desktop loaded, choose **Main Menu => System Settings => Printing** or type `redhat-config-printer` at a shell prompt.


Note

The **Printer Configuration Tool** saves any printer configurations made to the file `/etc/printcap`. If you have any printer configurations that you wish to add outside of the **Printer Configuration Tool**, you must add them to the file `/etc/printcap.local`, or it will be deleted whenever you run the tool or reboot your server.

To add a new printer:

1. Click the **New** button and type a queue name. The name should be descriptive enough to distinguish itself with other printers or network resources (such as **`workgroup_laser`** or **`third-floor_inkjet`**).

2. Choose a printer **Queue Type** from the list. Since you are configuring a print server, you will most likely choose a **Local Printer Device** since the printer will be attached physically to the server. For configuring JetDirect printers or printers attached to Windows (via SMB) or Netware (Novell) machines, refer to the chapter called "Printer Configuration" in the *Official Red Hat Linux Customization Guide*.
3. You have the option to select the **Printer Device** shown, rescan your devices for the correct device, or create your own custom device. This will be the `/dev` entry that represents the interface between the device driver and the printer itself. In most instances, the device will be named `/dev/lp0`.
4. Now you can choose a **Printer Driver** by selecting one from the extensive list. Choose the printer make and model, and select the driver. Drivers marked with an asterisk (*) are recommended drivers. If you are configuring a remote printer or a printer that does not have a corresponding driver for its make and model, the safest choice would be **Postscript Printer** (for JetDirect printers, **Raw Print Queue** is recommended).
5. Print a test page to make sure your printer is working properly.

Once the printer and print server is installed and configured properly, you should configure client computers to access the print server. The instructions are the same as those to configure the printer on the print server; however, instead of a local printer, you should choose a **UNIX Printer** queue type and type the print server hostname and port (usually 631) into the appropriate fields.Clients should now be able to begin printing.

Any post-install configuration on printers or the print server can be done via the Printer Configuration Tool by clicking **Edit** and choosing the **Driver Options**.

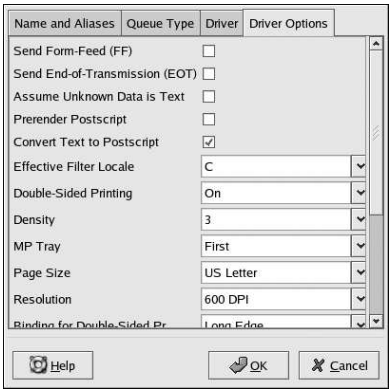

Figure 7-2. Driver Options

You can configure printer elements such as paper size, Postscript pre-rendering in the print driver itself (to deal with international glyphs such as Asian characters on a non-Asian character printer), page size, and more. For more information on driver options, refer to the *Official Red Hat Linux Customization Guide*.

7.9. Printer Sharing and Access Control

Now that your printer(s) are configured and ready for use, you can now begin to further customize your print servers to share to clients. By default, all clients on your network will be able send requests and

jobs to your new print server. Moreover, since modern print servers use the Internet Printing Protocol, any requests from the *Internet* are accepted as well, which can become a security issue. You should have your firewall configured to block port 631, on which the Red Hat Linux printing system listens. You can further configure the server to restrict access of printers to certain users or groups. Such restrictions on resources is termed *access control*. Red Hat Linux has several facilities for restricting access to server resources.

7.9.1. Printer Sharing with LPRng and the `/etc/hosts.lpd` File

For pure Linux or Linux/UNIX environments, printer sharing can be controlled using the `/etc/hosts.lpd` file. This file is not created by default; as root, create the file `/etc/hosts.lpd` on the machine to which the printer is attached. On separate lines in the file, add the IP address or hostname of each machine which should have printing privileges:

```
falcon.example.com
pinky.example.com
samiam.example.com
pigdog.example.com
yeti.example.com
```

Finally, restart the `lpd` printer daemon by issuing the command `/sbin/service lpd restart` (as root).

7.9.2. Printer Sharing with CUPS and `lpadmin`

Printer sharing for Linux/UNIX environments can also be controlled using the `lpadmin` command.

Because `lpadmin` is part of CUPS, you must first ensure that your system is configured to use CUPS as the default printing system. To do this, launch the **Printer System Switcher** application by executing the command `redhat-switch-printer` and selecting **CUPS**.

Once CUPS has been selected as the default printing system, you can then use `lpadmin` to make the necessary change. For example, to allow only a few select users to use your expensive graphical inkjet printer, run the following command:

```
lpadmin -p graphic_inkjet -u allow:bob,ted,alice
```

Note that *only* the users you specify will be able to print to the `graphic_inkjet` printer. Root and other users will not be able to access it. The resulting entry will be added to `/etc/cups/printers.conf`:

```
<Printer graphic_inkjet>
  Info This printer was modified by the lpadmin command
  Location Dustbin or Black Hole
  DeviceURI file:/dev/null
  State Idle
  Accepting Yes
  JobSheets none none
  AllowUser bob
  AllowUser ted
  AllowUser alice
</Printer>
```

You can edit this file in a text editor to modify or add elements to the access control list, such as hostnames. For more information on using `lpadmin`, type `man lpadmin` at a shell prompt.

7.9.3. Printer Sharing with Samba

If you are setting up a print server in a heterogeneous environment where users run various operating systems (such as Linux and Windows), you can restrict printer access using Samba. The central point for configuration of Samba services (including file sharing and administration) is the file `/etc/samba/smb.conf`. The following is an example of setting up sharing for Windows clients and Linux clients using Samba.

```
# NOTE: If you have a BSD-style print system there is no need to
# specifically define each individual printer
[printers]
   comment = All Printers
   path = /var/spool/samba
   printer = raw
   browseable = no
# Set public = yes to allow user 'guest account' to print
   guest ok = no
   writable = no
   printable = yes
```

In the example above, browsing has been turned off, so clients must explicitly configure the device instead of being able to browse the printer via Windows **Network Neighborhood**. Set the flag to **`yes`** to allow browsing. Also, set `guest ok = no` to **`yes`** to allow guest machines to print to your print server. Since Windows users will probably use the Windows-supported print drivers and send the binary print jobs to the printer, the setting `printer = raw` is used so that the print server does not use its own filters on top of the Windows print driver filter, potentially corrupting the output.

To restrict certain users access to printing services, the `valid users` option should be added. For example, to allow print access only to user **`fred`** and the group **`@design`**, set `guest ok` to **`no`** and add the following line:

```
valid users = fred @design
```

If your samba service is already started or running, you should restart it each time you edit the `/etc/samba/smb.conf` file by typing `/sbin/service smb restart` at a shell prompt.

7.10. Additional Resources

Printing configuration and network printing is a broad topic requiring knowledge and experience with both hardware, networking, and system administration. For more detailed information about deploying printer services in your environments, refer to following resources.

7.10.1. Installed Documentation

- `lpadmin` man page — Configure Red Hat Linux print services with this utility.
- `smb.conf` man page — Configure printer sharing using SMB/CIFS with this configuration file.
- `/usr/share/doc/cups-<version>` — The document collection for the Common UNIX Printing System (CUPS), an Internet Printing Protocol-compliant system for UNIX and Linux.

7.10.2. Useful Websites

- http://www.webopedia.com/TERM/p/printer.html — General definition of printers and descriptions of printer types.
- http://www.linuxprinting.org — A database of documents about printing, along with a database of nearly 1000 printers compatible with Linux printing facilities.
- http://www.tldp.org/HOWTO/Printing-HOWTO/index.html — *The Linux Printing-HOWTO* from the Linux Documentation Project.

7.10.3. Related Books

- *Network Printing* by Matthew Gast and Todd Radermacher; O'Reilly & Associates, Inc. — Comprehensive information on using Linux as a print server in heterogeneous environments.
- The *Official Red Hat Linux Customization Guide* and *Official Red Hat Linux Reference Guide* by Red Hat both cover networking and printing topics and are good sources of information if you run into any issues.

Thinking About the Unthinkable

redhat.

Chapter 8.

Planning for Disaster

Disaster planning is a subject that is easy for a system administrator to forget — it is not pleasant and it always seems that there is something else more pressing to do. However, letting disaster planning slide is one of the worst things you can do.

Although it is often the dramatic disasters (such as fire, flood, or storm) that first come to mind, the more mundane problems (such as construction workers cutting cables) can be just as disruptive. Therefore, the definition of a disaster that a system administrator should keep in mind is any unplanned event that disrupts the normal operation of your organization.

While it would be impossible to list all the different types of disasters that could strike, we will look at the leading factors that are part of each type of disaster. In this way, you can start looking at any possible exposure not in terms of its likelihood, but in terms of the factors that could cause that disaster.

8.1. Types of Disasters

In general, there are four different factors that can trigger a disaster. These factors are:

- Hardware failures
- Software failures
- Environmental failures
- Human errors

We will now look at each factor in more detail.

8.1.1. Hardware Failures

Hardware failures are easy to understand — the hardware fails, and work grinds to a halt. What is more difficult to understand is the nature of the failures, and how your exposure to them can be minimized. Here are some approaches that you can use:

8.1.1.1. Keeping Spare Hardware

At its simplest, exposure due to hardware failures can be reduced by having spare hardware available. Of course, this approach assumes two things:

- Someone on-site has the necessary skills to diagnose the problem, identify the failing hardware, and replace it.
- A replacement for the failing hardware is available.

These issues are covered in more detail below.

8.1.1.1.1. Having the Skills

Depending on your past experience and the hardware involved, having the necessary skills might be a non-issue. However, if you have not worked with hardware before, you might consider looking into local community colleges for introductory courses on PC repair. While such a course is not in and of itself sufficient to prepare you for tackling problems with an enterprise-level server, you will learn the basics (proper handling of tools and components, basic diagnostic procedures, and so on).


Tip

Before taking the approach of first fixing it yourself, make sure that the hardware in question:

- Is not still under warranty
- Is not under a service/maintenance contract of any kind

If you attempt repairs on hardware that is covered by a warranty and/or service contract, you are likely violating the terms of these agreements, and jeopardizing your continued coverage.

However, even with minimal skills, it might be possible to effectively diagnose and replace failing hardware — if you choose your stock of replacement hardware properly.

8.1.1.1.2. What to Stock?

This question illustrates the multi-faceted nature of anything related to disaster recovery. When considering what hardware to stock, here are some of the issues you should keep in mind:

- Maximum allowable downtime
- The skill required to affect a repair
- Budget available for spares
- Storage space required for spares
- Other hardware that could utilize the same spares

Each of these issues has a bearing on the types of spares that should be stocked. For example, stocking complete systems would tend to minimize downtime and require minimal skills to install, but would be much more expensive than having a spare CPU and RAM module on a shelf. However, this expense might be worthwhile if your organization has several dozen identical servers that could benefit from a single spare system.

No matter what the final decision, the following question is inevitable, and is discussed next.

8.1.1.1.2.1. How Much to Stock?

The question of spare stock levels is also multi-faceted. Here the main issues are:

- Maximum allowable downtime
- Projected rate of failure
- Estimated time to replenish stock
- Budget available for spares
- Storage space required for spares
- Other hardware that could utilize the same spares

At one extreme, for a system that can afford to be down a maximum of two days, and a spare that might be used once a year and could be replenished in a day, it would make sense to carry only one spare (and maybe even none, if you were confident of your ability to secure a spare within 24 hours).

At the other end of the spectrum, a system that could afford to be down no more than a few minutes, and a spare that might be used once a month (and could take several weeks to replenish) might mean that a half dozen spares (or more) should be on the shelf.

8.1.1.1.3. Spares That Are Not Spares

When is a spare not a spare? When it is hardware that is in day-to-day use, but is also available to serve as a spare for a higher-priority system should the need arise. This approach has some benefits:

- Less money dedicated to "non-productive" spares
- The hardware is known to be operative

There are, however, downsides to this approach:

- Normal production of the lower-priority task is interrupted
- There is an exposure should the lower-priority hardware fail (leaving no spare for the higher-priority hardware)

Given these constraints, the use of another production system as a spare may work, but the success of this approach will hinge on the system's specific workload, and how the system's absence will impact overall data center operations.

8.1.1.2. Service Contracts

Service contracts make the issue of hardware failures someone else's problem. All that is necessary for you to do is to confirm that a failure has, in fact, occurred and that it does not appear to have a software-related cause. You then make a telephone call, and someone shows up to make things right again.

It seems so simple. But as with most things in life, there is more to it than meets the eye. Here are some things that you will need to consider when looking at a service contract:

- Hours of coverage
- Response time
- Parts availability
- Available budget
- Hardware to be covered

We will explore each of these details more closely below.

8.1.1.2.1. Hours of Coverage

Different service contracts are available to meet different needs; one of the big variables between different contracts relates to the hours of coverage. Unless you are willing to pay a premium for the privilege, you cannot call just any time and expect to see a technician at your door a short time later.

Instead, depending on your contract, you might find that you cannot even phone the service company until a specific day/time, or if you can, they will not dispatch a technician until the day/time specified for your contract.

Most hours of coverage are defined in terms of the hours and the days during which a technician may be dispatched. Some of the more common coverage hours are:

- Monday through Friday, 9:00 to 17:00
- Monday through Friday, 12/18/24 hours each day (with the start and stop times mutually agreed upon)
- Monday through Saturday (or Monday through Sunday), same times as above

As you might expect, the cost of a contract increases with the hours of coverage. In general, extending the coverage Monday through Friday will cost less than adding on Saturday and Sunday coverage.

But even here there is a possibility to reduce costs if you are willing to do some of the work.

8.1.1.2.1.1. Depot Service

If your situation does not require anything more than the availability of a technician during standard business hours and you have sufficient experience to be able to determine what is broken, you might consider looking at *depot service*. Known by many names (including *walk-in service* and *drop-off service*), manufacturers may have service depots where technicians will work on hardware brought in by customers.

Depot service has the benefit of being as fast as you are. You do not have to wait for a technician to become available and show up at your facility. Depot technicians normally work at the depot full-time, meaning that there will be someone to work on your hardware as soon as you can get it to the depot.

Because depot service is done at a central location, there is a better chance that any parts that are required will be available. This can eliminate the need for an overnight shipment, or waiting for a part to be driven several hundred miles from another office that just happens to have that part in stock.

There are some trade-offs, however. The most obvious is that you cannot choose the hours of service — you get service when the depot is open. Another aspect to this is that the technicians will not work past their quitting time, so if your system failed at 16:30 on a Friday and you got the system to the depot by 17:00, it will likely not be worked on until the technicians arrive at work the following Monday morning.

Another trade-off is that depot service depends on having a depot nearby. If your organization is located in a metropolitan area, this is likely not going to be a problem. However, organizations in more rural locations will find that a depot may be a long drive away.

Tip

If considering depot service, take a moment and consider the mechanics of actually getting the hardware to the depot. Will you be using a company vehicle or your own? If your own, does your vehicle have the necessary space and load capacity? What about insurance? Will more than one person be necessary to load and unload the hardware?

Although these are rather mundane concerns, they should be addressed before making the decision to use depot service.

8.1.1.2.2. Response Time

In addition to the hours of coverage, many service agreements specify a level of response time. In other words, when you call requesting service, how long will it be before a technician arrives? As you might imagine, a faster response time equates to a more expensive service agreement.

There are limits to the response times that are available. For instance, the travel time from the manufacturer's office to your facility has a large bearing on the response times that are possible[1]. Response times in the four hour range are usually considered among the quicker offerings. Slower response

1. And this would likely be considered a best-case response time, as technicians usually are responsible for territories that extend away from their office in all directions. If you are at one end of their territory and the only available technician is at the other end, the response time will be even longer.

times can range from eight hours (which becomes effectively "next day" service for a standard business hours agreement), to 24 hours. As with every other aspect of a service agreement, even these times are negotiable — for the right price.

Note

Although it is not a common occurrence, you should be aware that service agreements with response time clauses can sometimes stretch a manufacturer's service organization beyond its ability to respond. It is not unheard of for a very busy service organization to send somebody — *anybody* — on a short response-time service call just to meet their response time commitment. This person can then appear to diagnose the problem, calling "the office" to have someone bring in "the right part".

In fact, they are just waiting for a technician that is actually capable of handling the call to arrive.

While it might be understandable to see this happen under extraordinary circumstances (such as power problems that have damaged systems throughout their service area), if this is a consistent method of operation you should contact the service manager, and demand an explanation.

If your response time needs are stringent (and your budget correspondingly large), there is one approach that can cut your response times even further — to zero.

8.1.1.2.2.1. Zero Response Time — Having an On-Site Technician

Given the appropriate situation (you are one of the biggest customers in the area), sufficient need (downtime of *any* magnitude is unacceptable), and financial resources (if you have to ask for the price, you probably cannot afford it), you might be a candidate for a full-time, on-site technician. The benefits of having a technician always standing by are obvious:

- Instant response to any problem
- A more proactive approach to system maintenance

As you might expect, this option can be *very* expensive, particularly if you require an on-site technician 24X7. But if this approach is appropriate for your organization, you should keep a number of points in mind in order to gain the most benefit.

First, an on-site technician will need many of the resources of a regular employee, such as a workspace, telephone, appropriate access cards and/or keys, and so on.

On-site technicians are not very helpful if they do not have the proper parts. Therefore, make sure that secure storage is set aside for the technician's spare parts. In addition, make sure that the technician keeps a stock of parts appropriate for your configuration, and that those parts are not "cannibalized" by other technicians for their customers.

8.1.1.2.3. Parts Availability

Obviously, the availability of parts plays a large role in limiting your organization's exposure to hardware failures. In the context of a service agreement, the availability of parts takes on another dimension, as the availability of parts applies not only to your organization, but to any other customer in the manufacturer's territory that might need those parts as well. Another organization that has purchased more of the manufacturer's hardware than you might get preferential treatment when it comes to getting parts (and technicians, for that matter).

Unfortunately, there is little that can be done in such circumstances, short of working out the problem with the service manager.

8.1.1.2.4. Available Budget

As outlined above, service contracts vary in price according to the nature of the services being provided. Keep in mind that the costs associated with a service contract are a recurring expense; each time the contract is due to expire you will need to negotiate a new contract and pay again.

8.1.1.2.5. Hardware to be Covered

Here is an area where you might be able to help keep costs to a minimum. Consider for a moment that you have negotiated a service agreement that has an on-site technician 24X7, on-site spares — you name it. Every single piece of hardware you have purchased from this vendor is covered, including the PC that the company receptionist uses to surf the Web while answering phones and handing out visitor badges.

Does that PC *really* need to have someone on-site 24X7? Even if the PC were vital to the receptionist's job, the receptionist only works from 9:00 to 17:00; it is highly unlikely that:

- The PC will be in use from 17:00 to 09:00 the next morning (not to mention weekends)
- A failure of this PC will be noticed, except between 09:00 and 17:00

Therefore, paying to enable having this PC serviced in the middle of a Saturday night is simply a waste of money.

The thing to do is to split up the service agreement such that non-critical hardware is grouped separately from the more critical hardware. In this way, costs can be kept as low as possible.

Note

If you have twenty identically-configured servers that are critical to your organization, you might be tempted to have a high-level service agreement written for only one or two, with the rest covered by a much less expensive agreement. Then, the reasoning goes, no matter which one of the servers fails on a weekend, you will say that *it* is the one eligible for high-level service.

Do not do this. Not only is it dishonest, most manufacturers keep track of such things by using serial numbers. Even if you figure out a way around such checks, you will spend far more after being discovered than you will by simply being honest and paying for the service you really need.

8.1.2. Software Failures

Software failures can result in extended downtimes. For example, customers of a highly-available computer system recently experienced this firsthand when a bug in the time handling code of the computer's operating system resulted in every customer's system crashing at a certain time of a certain day. While this particular situation is a more spectacular example of a software failure in action, other software-related failures may be less dramatic, but still as devastating.

Software failures can strike in one of two areas:

- In the operating system
- In applications

Each type of failure has its own specific impact; we will explore each in more detail below.

8.1.2.1. Operating System Failures

In this type of failure, the operating system is responsible for the failure. The type of failures come from two main areas:

- Crashes
- Hangs

The main thing to keep in mind about operating system failures is that they take out everything that the computer was running at the time of the failure. As such, operating system failures can be devastating to production.

8.1.2.1.1. Crashes

Crashes occur when the operating system experiences an error condition from which it cannot recover. The reasons for crashes can range from an inability to handle an underlying hardware problem, to a bug in kernel-level code. When an operating system crashes, the system must be rebooted in order to continue production.

8.1.2.1.2. Hangs

When the operating system stops handling system events, the system grinds to a halt. This is known as a *hang*. Hangs can be caused by *deadlocks* (two resource consumers contending for resources the other has) and *livelocks* (two or more processes responding to each other's activities, but doing no useful work), but the end result is the same — a complete lack of productivity.

8.1.2.2. Application Failures

Unlike operating system failures, application failures can be more limited in the scope of their damage. Depending on the specific application, a single application failing might impact only one person. On the other hand, if it is a server application servicing a large population of client applications, the failure could be much more widespread.

8.1.3. Environmental Failures

Even though the hardware is running perfectly, and even though the software is configured properly and is working as it should, problems can still occur. The most common problems that occur outside of the system itself have to do with the physical environment in which the system is running.

Environmental issues can be broken into four major categories:

- Building integrity
- Electricity
- Air conditioning
- Weather and the outside world

8.1.3.1. Building Integrity

For such a seemingly simple structure, a building performs a great many functions. It provides shelter from the elements. It provides the proper micro-climate for the building's contents. It has mechanisms to provide power and to protect against fire and theft/vandalism. Performing all these functions, it is not surprising that there is a great deal that can go wrong with a building. Here are some possibilities to consider:

- Roofs can leak into data centers.
- Various building systems (such as water, sewer, or air handling) can fail, rendering the building uninhabitable.
- Floors may have insufficient load-bearing capacity to hold everything you want to put in the data center.

It is important to have a creative mind when it comes to thinking about the different ways buildings can fail. The list above is only meant to start you thinking along the proper lines.

8.1.3.2. Electricity

Because electricity is the lifeblood of any computer system, power-related issues are paramount in the mind of system administrators everywhere. There are several different aspects to power; we will cover them in more detail below.

8.1.3.2.1. The Security of Your Power

First, it is necessary to determine how secure your normal power supply may be. Just like nearly every other data center, you probably obtain your power from a local power company via power transmission lines. Because of this, there are limits to what you can do to make sure that your primary power supply is as secure as possible.

Tip

Organizations located near the boundaries of a power company might be able to negotiate connections to two different power grids:

- The one servicing your area
- The one from the neighboring power company

The costs involved in running power lines from the neighboring grid are sizable, making this an option only for larger organizations. However, such organizations find that the redundancy gained outweigh the costs in many cases.

The main things to check are the methods by which the power is brought onto your organization's property and into the building. Are the transmission lines above ground or below? Above-ground lines are susceptible to:

- Damage from extreme weather conditions (ice, wind, lightning)
- Traffic accidents that damage the poles and/or transformers
- Animals straying into the wrong place and shorting out the lines

Below-ground lines have their own unique shortcomings:

- Damage from construction workers digging in the wrong place
- Flooding
- Lightning (though much less so than above-ground lines)

Continue to trace the power lines into your building. Do they first go to an outside transformer? Is that transformer protected from vehicles backing into it or trees falling on it? Are all exposed shutoff switches locked?

Once inside your building, could the power lines (or the panels to which they attach) be subject to other problems? For instance, could a plumbing problem flood the electrical room?

Continue tracing the power into the data center; is there anything else that could unexpectedly interrupt your power supply? For example, is the data center sharing a circuit with non-data center loads? If so, the external load might one day trip the circuit's overload protection, taking down the data center as well.

8.1.3.2.2. Power Quality

It is not enough to ensure that the data center's power source is as secure as possible. You must also be concerned with the quality of the power being distributed throughout the data center. There are several factors that must be considered:

Voltage

> The voltage of the incoming power must be stable, with no voltage reductions (often called *sags*, *droops*, or *brownouts*) or voltage increases (often known as *spikes* and *surges*).

Waveform

> The waveform must be a clean sine wave, with minimal *THD* (Total Harmonic Distortion).

Frequency

> The frequency must be stable (most countries use a power frequency of either 50Hz or 60Hz).

Noise

> The power must not include any *RFI* (Radio Frequency Interference) or *EMI* (Electro-Magnetic Interference) noise.

Current

> The power must be supplied at a current rating sufficient to run the data center.

Power supplied directly from the power company will not normally meet the standards necessary for a data center. Therefore, some level of power conditioning is usually required. There are several different approaches possible:

Surge Protectors

> Surge protectors do just what their name implies — they filter surges from the power supply. Most do nothing else, leaving equipment vulnerable to damage from other power-related problems.

Power Conditioners

> Power conditioners attempt a more comprehensive approach; depending on the sophistication of the unit, power conditioners often can take care of most of the types of problems outlined above.

Motor-Generator Sets

A motor-generator set is essentially a large electric motor powered by your normal power supply. The motor is attached to a large flywheel, which is, in turn, attached to a generator. The motor turns the flywheel and generator, which generates electricity in sufficient quantities to run the data center. In this way, the data center power is electrically isolated from outside power, meaning that most power-related problems are eliminated. The flywheel also provides the ability to maintain power through short outages, as it takes several seconds for the flywheel to slow to the point at which it can no longer generate power.

Uninterruptible Power Supplies

Some types of Uninterruptible Power Supplies (more commonly known as *UPSs*) include most (if not all) of the protection features of a power conditioner[2].

With the last two technologies listed above, we have started in on the topic most people think of when they think about power — backup power. In the next section, we will look at different approaches to providing backup power.

8.1.3.2.3. Backup Power

One power-related term that nearly everyone has heard is the term *blackout*. A blackout is a complete loss of electrical power, and may last from a fraction of a second to weeks.

Because the length of blackouts can vary so greatly, it is necessary to approach the task of providing backup power using different technologies for different lengths of blackouts.

Tip

The most frequent blackouts last, on average, no more than a few seconds; longer outages are much less frequent. Therefore, concentrate first on protecting against blackouts of only a few minutes in length, then work out methods of reducing your exposure to longer outages.

8.1.3.2.3.1. Providing Power For the Next Few Seconds

Since the majority of outages last only a few seconds, your backup power solution must have two primary characteristics:

- Very short time to switch to backup power (known as *transfer time*)
- A *runtime* (the time that backup power will last) measured in seconds to minutes

The backup power solutions that match these characteristics are motor-generator sets and UPSs. The flywheel in the motor-generator set allows the generator to continue producing electricity for enough time to ride out outages of a few seconds. Motor-generator sets tend to be quite large and expensive, making them a practical solution for mid-sized and larger data centers.

However, another technology can fill in for those situations where a motor-generator set is too expensive, as well as handling longer outages.

2. We will discuss UPSs in more detail in Section 8.1.3.2.3.2.

8.1.3.2.3.2. Providing Power For the Next Few Minutes

UPSs can be purchased in a variety of sizes — small enough to run a single low-end PC for five minutes, or large enough to power an entire data center for an hour or more.

UPSs are made up of the following parts:

- A *transfer switch* for switching from the primary power supply to the backup power supply
- A battery, for providing backup power
- An *inverter*, which converts the DC current from the battery into the AC current required by the data center hardware

Apart from the size and battery capacity of the unit, UPSs come in two basic types:

- The *offline* UPS uses its inverter to generate power only when the primary power supply fails.
- The *online* UPS uses its inverter to generate power all the time, powering the inverter via its battery only when the primary power supply fails.

Each type has their advantages and disadvantages. The offline UPS is usually less expensive, because the inverter does not have to be constructed for full-time operation. However, a problem in the inverter of an offline UPS will go unnoticed (until the next power outage, that is).

online UPSs tend to be better at providing clean power to your data center; after all, an online UPS is essentially generating power for you full time.

But no matter what type of UPS you choose, you must properly size the UPS to your anticipated load (thereby ensuring that the UPS has sufficient capacity to produce electricity at the required voltage and current), *and* you must determine how long you would like to be able to run your data center on battery power.

To determine this information, you must first identify those loads that will be serviced by the UPS. Go to each piece of equipment and determine how much power it draws (this is normally listed near the unit's power cord). Write down the voltage, watts, and/or amps. Once you have these figures for all of the hardware, you will need to convert them to *VA* (Volt-Amps). If you have a wattage number, you can simply use the listed wattage as the VA; if you have amps, multiply it by volts to get VA. By adding the VA figures you can arrive at the approximate VA rating required for the UPS.

Note

Strictly speaking, this approach to calculating VA is not entirely correct; however, to get the true VA you would need to know the power factor for each unit, and this information is rarely, if ever, provided. In any case, the VA numbers you will obtain from this approach will reflect worst-case values, leaving a large margin of error for safety.

Determining runtime is more of a business question than a technical question — what sorts of outages are you willing to protect against, and how much money are you prepared to spend to do so? Most sites select runtimes that are less than an hour or two at most, as battery-backed power becomes very expensive beyond this point.

8.1.3.2.3.3. Providing Power For the Next Few Hours (and Beyond)

Once we get into power outages that are measured in days, the choices get even more expensive. At this point the technologies are limited to generators powered by some type of engine — diesel and gas turbine, primarily.

At this point, your options are wide open, assuming your organization has sufficient funds. This is also an area where experts should help you determine the best solution for your organization. Very few system administrators will have the specialized knowledge to plan the acquisition and deployment of these kinds of power generation systems.


Tip

Portable generators of all sizes can be rented, making it possible to have the benefits of generator power without the initial outlay of money necessary to purchase one. However, keep in mind that in disasters affecting your general vicinity, rented generators will be in very short supply and very expensive.

8.1.3.2.4. Planning for Extended Outages

While a black out of five minutes is little more than an inconvenience to the personnel in a darkened office, what about an outage that lasts an hour? Five hours? A day? A week?

The fact is, at some point even if the data center is operating normally, an extended outage will eventually affect your organization. Consider the following points:

- What if there is no power to maintain environmental control in the data center?
- What if there is no power to maintain environmental control in the entire building?
- What if there is no power to operate personal workstations, the telephone system, the lights?

The point here is that your organization will need to determine at what point an extended outage will just have to be tolerated. Or if that is not an option, your organization will need to reconsider its ability to function completely independently of on-site power for extended periods, meaning that very large generators will be needed to power the entire building.

Of course, even this level of planning cannot take place in a vacuum. It is very likely that whatever caused the extended outage is likely affecting the world outside of your organization, and that the outside world will start having an affect on your organization's ability to continue operations, even given unlimited power generation capacity.

8.1.3.3. Heating, Ventilation, and Air Conditioning

Heating, Ventilation, and Air Conditioning (*HVAC*) systems used in today's office buildings are incredibly sophisticated. Often computer controlled, the HVAC system is vital to providing a comfortable work environment.

Data centers usually have additional air handling equipment, primarily to remove the heat generated by the many computers and associated equipment. Failures in an HVAC system can be devastating to the continued operation of a data center. And given their complexity and electro-mechanical nature, the possibilities for failure are many and varied. Here are a few examples:

- The air handling units (essentially large fans driven by large electric motors) can fail due to electrical overload, bearing failure, belt/pulley failure, etc.
- The cooling units (often called *chillers*) can lose their refrigerant due to leaks, or can have their compressors and/or motors seize.

Again, HVAC repair and maintenance is a very specialized field that the average system administrator should leave to the experts. If anything, a system administrator should make sure that the HVAC equipment serving the data center is checked for normal operation on a daily basis (if not more frequently), and is maintained according to the manufacturer's guidelines.

8.1.3.4. Weather and the Outside World

There are some types of weather that will obviously cause problems for a system administrator:

- Heavy snow and ice can prevent personnel from getting to the data center, and can even clog air conditioning condensers, resulting in elevated data center temperatures just when no one is able to get to the data center to take corrective action.
- High winds can disrupt power and communications, with extremely high winds actually doing damage to the building itself.

There are other types of weather than can still cause problems, even if they are not as well known. For example, exceedingly high temperatures can result in overburdened cooling systems, and brownouts or blackouts as the local power grid becomes overloaded.

Although there is little that can be done about the weather, knowing the way that it can affect your data center operations can help you to keep running even when the weather turns bad.

8.1.4. Human Errors

It has been said that computers really *are* perfect. The reasoning is that if you dig deeply enough, behind every computer error you will find the human error that caused it. In this section, we will explore the more common types of human errors and their impacts.

8.1.4.1. System Administrator Errors

System administrators sometimes make unnecessary work for themselves when they are not careful about what they are doing. During the course of carrying out day-to-day responsibilities, system administrators have more than sufficient access to the computer systems (not to mention their super-user access privileges) to mistakenly bring systems down.

System administrators either make errors of misconfiguration, or errors during maintenance.

8.1.4.1.1. Misconfiguration Errors

System administrators must often configure various aspects of a computer system. This configuration might include:

- Email
- User accounts
- Network
- Applications

The list could go on quite a bit longer. The actual task of configuration varies greatly; some tasks require editing a text file (using any one of a hundred different configuration file syntaxes), while other tasks require running a configuration utility.

The fact that these tasks are all handled differently is merely an additional challenge to the basic fact that each configuration task itself requires different knowledge. For example, the knowledge re-

quired to configure a mail transport agent is fundamentally different from the knowledge required to configure a new network connection.

Given all this, perhaps it should be surprising that so *few* mistakes are actually made. In any case, configuration is, and will continue to be, a challenge for system administrators. Is there anything that can be done to make the process less error-prone?

8.1.4.1.1.1. Change Control

The common thread of every configuration change is that some sort of a change is being made. The change may be large, or it may be small. But it is still a change, and should be treated in a particular way.

Many organizations implement some type of a change control process. The intent is to help system administrators (and all parties affected by the change) to manage the process of change, and to reduce the organization's exposure to any errors that may occur.

A change control process normally breaks the change into different steps. Here is an example:

Preliminary research

Preliminary research attempts to clearly define:

- The nature of the change to take place.
- Its impact, should the change the succeed.
- A fallback position, should the change fail.
- An assessment of what types of failures are possible.

Preliminary research might include testing the proposed change during a scheduled downtime, or it may go so far as to include implementing the change first on a special test environment run on dedicated test hardware.

Scheduling

Here, the change is examined with an eye toward the actual mechanics of implementation. The scheduling being done here includes outlining the sequencing and timing of the change (along with the sequencing and timing of any steps necessary to back the change out should a problem arise), as well as ensuring that the time allotted for the change is sufficient and does not conflict with any other system-level activity.

The product of this process is often a checklist of steps for the system administrator to use while making the change. Included with each step are instructions to perform in order to back out the change should the step fail. Estimated times are often included, making it easier for the system administrator to determine whether the work is on schedule or not.

Execution

At this point, the actual execution of the steps necessary to implement the change should be straightforward and anti-climactic. The change is either implemented, or (if trouble crops up) it is backed out.

Monitoring

Whether the change is implemented or not, the environment is monitored to make sure that everything is operating as it should.

Documenting

If the change has been implemented, all existing documentation is updated to reflect the changed configuration.

Obviously, not all configuration changes require this level of detail. Creating a new user account should not require any preliminary research, and scheduling would likely consist of determining whether the system administrator has a spare moment to create the account. Execution would be similarly quick; monitoring might consist of ensuring that the account was usable, and documenting would probably entail sending an email to the user's manager.

But as the configuration changes become more complex, a more formal change control process becomes necessary.

8.1.4.1.2. Mistakes Made During Maintenance

This type of error can be insidious because there is usually so little planning and tracking done during day-to-day maintenance. System administrators see the results of this kind of error every day, especially from the many users that swear they did not change a thing — the computer just broke. The user that says this usually does not remember what they did, and when the same thing happens to you, you will probably not remember what you did, either.

The key thing to keep in mind is that you must be able to remember what changes you made during maintenance if you are to be able to resolve any problems quickly. A full-blown change control process is simply not realistic for the hundreds of small things done over the course of a day. What can be done to keep track of the 101 small things a system administrator does every day?

The answer is simple — takes notes. Whether it is done in a notebook, a PDA, or as comments in the affected files, take notes. By tracking what you have done, you will stand a better chance of seeing a failure as being related to a change you recently made.

8.1.4.2. Operations Personnel Errors

Operators have a different relationship with an organization's computers than system administrators. Operators tend to have a more formal tie to the computers, using them in ways that have been dictated by others. Therefore, the types of errors that an operator might make differ from those a system administrator might make.

8.1.4.2.1. Failure to Follow Procedures

Operators should have sets of procedures documented and available for nearly every action they perform[3]. It might be that an operator does not follow the procedures as they are laid out. There might be several reasons for this:

- The environment was changed at some time in the past, and the procedures were never updated. Now the environment changes again, rendering the operator's memorized procedure invalid. At this point, even if the procedures were updated (which is unlikely, given the fact that they were not updated before) this operator will not be aware of it.
- The environment was changed, and no procedures exist. This is just a more out-of-control version of the previous situation.
- The procedures exist and are correct, but the operator will not (or cannot) follow them.

3. If the operators at your organization do not have a set of operating procedures, work with them, your management, and your users to get them created. Without them, a data center is out of control, and likely to experience severe problems in the course of day-to-day operations.

Depending on the management structure of your organization, you might not be able to do much more than communicate your concerns to the appropriate manager. In any case, making yourself available to do what you can to help resolve the problem is the best approach.

8.1.4.2.2. Mistakes Made During Procedures

Even if the operator follows the procedures, and even if the procedures are correct, it is still possible for mistakes to be made. If this happens, the possibility exists that the operator is careless (in which case the operator's management should become involved).

Another explanation is that it was just a mistake. In these cases, the best operators will realize that something is wrong and seek assistance. Always encourage the operators you work with to contact the appropriate people immediately if they suspect something is wrong. Although many operators are highly-skilled and able to resolve many problems independently, the fact of the matter is that this is not their job. And a problem that is made worse by a well-meaning operator will harm both that person's career, and your ability to quickly resolve what might originally have been a small problem.

8.1.4.3. Service Technician Errors

Sometimes the very people that are supposed to help you keep your systems reliably running can actually make things worse. This is not due to any conspiracy; it is simply that anyone working on any technology for any reason risks rendering that technology inoperable. The same effect is at work when programmers fix one bug, but end up creating another.

8.1.4.3.1. Improperly-Repaired Hardware

In this case, the technician either failed to correctly diagnose the problem and made an unnecessary (and useless) repair, or the diagnosis was correct, but the repair was not carried out properly. It may be that the replacement part was itself defective, or that the proper procedure was not followed when the repair was carried out.

This is why it is important to be aware of what the technician is doing at all times. By doing this, you can keep an eye out for failures that seem to be related to the original problem in some way. This will keep the technician on track should there be a problem; otherwise there is a chance that the technician will view this fault as being new and unrelated to the one that was supposedly fixed. In this way, time will not be wasted chasing the wrong problem.

8.1.4.3.2. Fixing One Thing and Breaking Another

Sometimes, even though a problem was diagnosed and repaired successfully, another problem pops up to take its place. The CPU module was replaced, but the anti-static bag it came in was left in the cabinet, blocking the fan and causing an over-temperature shutdown. Or the failing disk drive in the RAID array was replaced, but because a connector on another drive was bumped and accidentally disconnected, the array is still down.

These things might be the result of chronic carelessness, or an honest mistake. It does not matter. What you should always do is to carefully review the repairs made by the technician and ensure that the system is working properly before letting the technician leave.

8.1.4.4. End-User Errors

The users of a computer can also make mistakes that can have serious impacts. However, due to their normally unprivileged operating environment, user errors tend to be errors that are more localized.

8.1.4.4.1. Improper Use of Applications

When applications are used improperly, various problems can occur:

- Files inadvertently overwritten
- Wrong data used as input to an application
- Files not clearly named and organized
- Files accidentally deleted

The list could go on, but this is enough to illustrate the point. Due to users not having super-user privileges, the mistakes they make are usually limited to their own files. As such, the best approach is two-pronged:

- Educate users in the proper use of their applications and in proper file management techniques
- Make sure backups of users' files are made regularly, and that the restoration process is as streamlined and quick as possible

Beyond this, there is little that can be done to keep user errors to a minimum.

8.2. Backups

Backups have two major purposes:

- To permit restoration of individual files
- To permit wholesale restoration of entire file systems

The first purpose is the basis for the typical file restoration request: a user accidentally deletes a file, and asks that it be restored from the latest backup. The exact circumstances may vary somewhat, but this is the most common day-to-day use for backups.

The second situation is a system administrator's worst nightmare: for whatever reason, the system administrator is staring at hardware that used to be a productive part of the data center. Now, it is little more than a lifeless chunk of steel and silicon. The thing that is missing is all the software and data you and your users have assembled over the years. Supposedly everything has been backed up. The question is: has it?

And if it has, will you be able to restore it?

8.2.1. Different Data: Different Backup Needs

If you look at the kinds of data[4] processed and stored by a typical computer system, you will find that some of the data hardly ever changes, and some of the data is constantly changing.

The pace at which data changes is crucial to the design of a backup procedure. There are two reasons for this:

- A backup is nothing more than a snapshot of the data being backed up. It is a reflection of that data at a particular moment in time.

4. We are using the term *data* in this section to describe anything that is processed via backup software. This includes operating system software, application software, as well as actual data. No matter what it is, as far as backup software is concerned, it is all simply data.

- Data that changes infrequently can be backed up infrequently; data that changes more frequently must be backed up more frequently.

System administrators that have a good understanding of their systems, users, and applications should be able to quickly group the data on their systems into different categories. However, here are some examples to get you started:

Operating System

This data only changes during upgrades, the installation of bug-fixes, and any site-specific modifications.


Tip

Should you even bother with operating system backups? This is a question that many system administrators have pondered over the years. On the one hand, if the installation process is relatively easy, the application of bug-fixes and customizations are well documented and easily reproducible, simply reinstalling the operating system may be a viable option.

On the other hand, if there is the least doubt that a fresh installation can completely recreate the original system environment, backing up the operating system is the best choice.

Application Software

This data changes whenever applications are installed, upgraded, or removed.

Application Data

This data changes as frequently as the associated applications are run. Depending on the specific application and your organization, this could mean that changes take place second-by-second, or once at the end of each fiscal year.

User Data

This data changes according to the usage patterns of your user community. In most organizations, this means that changes take place all the time.

Based on these categories (and any additional ones that are specific to your organization), you should have a pretty good idea concerning the nature of the backups that are needed to protect your data.


Note

You should keep in mind that most backup software deals with data on a directory or file system level. In other words, your system's directory structure will play a part in how backups will be performed. This is another reason why it is always a good idea to carefully consider the best directory structure for a new system, grouping files and directories according to their anticipated usage.

8.2.2. Backup Technologies

Red Hat Linux comes with several different programs for backing up and restoring data. By themselves, these utility programs do not constitute a complete backup solution. However, they can be used as the nucleus of such a solution, and as such, warrant some attention.

8.2.2.1. tar

The tar utility is well known among UNIX system administrators. It is the archiving method of choice for sharing ad-hoc bits of source code and files between systems. The tar implementation included with Red Hat Linux is GNU tar, one of the more feature-rich tar implementations.

Backing up the contents of a directory can be as simple as issuing a command similar to the following:

```
tar cf /mnt/backup/home-backup.tar /home/
```

This command will create an archive called home-backup.tar in /mnt/backup/. The archive will contain the contents of the /home/ directory. The archive file can be compressed by adding a single option:

```
tar czf /mnt/backup/home-backup.tar.gz /home/
```

The home-backup.tar.gz file is now gzip compressed.

There are many other options to tar; to learn more about them, read the tar man page.

8.2.2.2. cpio

The cpio utility is another traditional UNIX program. It is an excellent general-purpose program for moving data from one place to another and, as such, can serve well as a backup program.

The behavior of cpio is a bit different from tar. Unlike tar, cpio reads the files it is to process via standard input. A common method of generating a list of files for cpio is to use programs such as find whose output is then piped to cpio:

```
find /home | cpio -o > /mnt/backup/home-backup.cpio
```

This command creates a cpio archive called home-backup.cpio in the /mnt/backup directory.

There are many other options to cpio; to learn more about them see the cpio man page.

8.2.2.3. dump/restore: Not Recommended!

The dump and restore programs are Linux equivalents to the UNIX programs of the same name. As such, many system administrators with UNIX experience may feel that dump and restore are viable candidates for a good backup program under Red Hat Linux. Unfortunately, the design of the Linux kernel has moved ahead of dump's design. Here is Linus Torvald's comment on the subject:

```
From:  Linus Torvalds
To:  Neil Conway
Subject: Re: [PATCH] SMP race in ext2 - metadata corruption.
Date:  Fri, 27 Apr 2001 09:59:46 -0700 (PDT)
Cc:  Kernel Mailing List <linux-kernel At vger Dot kernel Dot org>

[ linux-kernel added back as a cc ]

On Fri, 27 Apr 2001, Neil Conway wrote:
> > I'm surprised that dump is deprecated (by you at least ;-)).  What to
> use instead for backups on machines that can't umount disks regularly?


Note that dump simply won't work reliably at all even in 2.4.x: the buffer
cache and the page cache (where all the actual data is) are not
coherent. This is only going to get even worse in 2.5.x, when the
directories are moved into the page cache as well.
```

```
So anybody who depends on "dump" getting backups right is already playing
Russian roulette with their backups.  It's not at all guaranteed to get the
right results - you may end up having stale data in the buffer cache that
ends up being "backed up".

Dump was a stupid program in the first place. Leave it behind.

> I've always thought "tar" was a bit undesirable (updates atimes or
> ctimes for example).

Right now, the cpio/tar/xxx solutions are definitely the best ones, and
will work on multiple filesystems (another limitation of "dump"). Whatever
problems they have, they are still better than the _guaranteed_(*)  data
corruptions of "dump".

However, it may be that in the long run it would be advantageous to have a
"filesystem maintenance interface" for doing things like backups and
defragmentation..

  Linus

(*) Dump may work fine for you a thousand times. But it _will_ fail under
the right circumstances. And there is nothing you can do about it.
```

Given this problem, the use of `dump`/`restore` is strongly discouraged.

8.2.3. Backup Software: Buy Versus Build

Now that we have seen the basic utility programs that do the actual work of backing up data, the next step is to determine how to integrate these programs into an overall process that does the following things:

- Schedules backups to run at the proper time
- Manages the location, rotation, and usage of backup media
- Works with operators (and/or robotic media changers) to ensure that the proper media is available
- Assists operators in locating the media containing a specific backup of a specific file

As you can see, a real-world backup solution entails much more than just typing a `tar` command.

Most system administrators at this point look at one of two solutions:

- Create an in-house developed backup system from scratch
- Purchase a commercially-developed solution

Each approach has its good and bad points. Given the complexity of the task, an in-house solution is not likely to handle some aspects (most notably media management) very well. However, for some organizations, this might not be a shortcoming.

A commercially-developed solution is more likely to be highly functional, but may also be overly-complex for the organization's present needs. That said, the complexity might make it possible to stick with one solution even as the organization grows.

As you can see, there is no clear-cut method for deciding on a backup system. The only guidance that can be offered is to ask you to consider these points:

- Changing backup software is difficult; once implemented, you will be using the backup software for a long time. After all, you will have long-term archive backups that you will need to be able to read. Changing backup software means you must either keep the original software around, or you must convert your archive backups to be compatible with the new software.
- The software must be 100% reliable when it comes to backing up what it is supposed to, when it is supposed to.
- When the time comes to restore any data — whether a single file, or an entire file system — the backup software must be 100% reliable.

Although this section has dealt with a build-or-buy decision, there is, in fact, another approach. There are open source alternatives available, and one of them is included with Red Hat Linux.

8.2.3.1. The Advanced Maryland Automatic Network Disk Archiver (AMANDA)

AMANDA is a client/server based backup application produced by the University of Maryland. By having a client/server architecture, a single backup server (normally a fairly powerful system with a great deal of free space on fast disks, and configured with the desired backup device) can back up many client systems, which need nothing more than the AMANDA client software.

This approach to backups makes a great deal of sense, as it concentrates those resources needed for backups in one system, instead of requiring additional hardware for every system requiring backup services. AMANDA's design also serves to centralize the administration of backups, making the system administrator's life that much easier.

The AMANDA server manages a pool of backup media, and rotates usage through the pool in order to ensure that all backups are retained for the administrator-dictated timeframe. All media is preformatted with data that allows AMANDA to detect whether the proper media is available or not. In addition, AMANDA can be interfaced with robotic media changing units, making it possible to completely automate backups.

AMANDA can use either `tar` or `dump` to do the actual backups (although under Red Hat Linux using `tar` is preferable, due to the issues with `dump` raised in Section 8.2.2.3). As such, AMANDA backups do not require AMANDA in order to restore files — a decided plus.

In operation, AMANDA is normally scheduled to run once a day during the data center's backup window. The AMANDA server connects to the client systems, and directs the clients to produce estimated sizes of the backups to be done. Once all the estimates are available, the server constructs a schedule, automatically determining the order in which systems will be backed up.

Once the backups actually start, the data is sent over the network from the client to the server, where it is stored on a holding disk. Once a backup is complete, the server starts writing it out from the holding disk to the backup media. At the same time, other clients are sending their backups to the server for storage on the holding disk. This results in a continuous stream of data available for writing to the backup media. As backups are written to the backup media, they are deleted from the server's holding disk.

Once all backups have been completed, the system administrator is emailed a report outlining the status of the backups, making review easy and fast.

Should it be necessary to restore data, AMANDA contains a utility program that allows the operator to identify the file system, date, and file name(s). Once this is done, AMANDA identifies the correct backup media, accesses, and restores the desired data. As stated earlier, AMANDA's design also makes it possible to restore data even without AMANDA's assistance, although identification of the correct media would be a slower, manual process.

This section has only touched upon the most basic AMANDA concepts. If you would like to do more research on AMANDA, your Red Hat Linux system has additional information. To learn more, type the following command for a list of documentation files available for AMANDA:

```
rpm -qd amanda-server
```

(Note that this command will only work if you have installed the `amanda` RPMs on your Red Hat Linux system.)

You can also learn more about AMANDA from the AMANDA website at http://www.amanda.org/.

8.2.4. Types of Backups

If you were to ask a person that was not familiar with computer backups, most would think that a backup was simply an identical copy of the data on the computer. In other words, if a backup was created Tuesday evening, and nothing changed on the computer all day Wednesday, the backup created Wednesday evening would be identical to the one created on Tuesday.

While it is possible to configure backups in this way, it is likely that you would not. To understand more about this, we first need to understand the different types of backups that can be created. They are:

- Full backups
- Incremental backups
- Differential backups

8.2.4.1. Full Backups

The type of backup that was discussed at the beginning of this section is known as a *full backup*. A full backup is simply a backup where every single file is written to the backup media. As noted above, if the data being backed up never changes, every full backup being created will be the same.

That similarity is due to the fact that a full backup does not check to see if a file has changed since the last backup; it blindly writes it to the backup media whether it has been modified or not.

This is the reason why full backups are not done all the time — every file is written to the backup media. This means that a great deal of backup media is used even if nothing has changed. Backing up 100 gigabytes of data each night when maybe 10 megabytes worth of data has changed is not a sound approach; that is why *incremental backups* were created.

8.2.4.2. Incremental Backups

Unlike full backups, incremental backups first look to see whether a file's modification time is more recent than its last backup time. If it is not, that file has not been modified since the last backup and can be skipped this time. On the other hand, if the modification date *is* more recent than the last backup date, the file has been modified and should be backed up.

Incremental backups are used in conjunction with an occasional full backup (for example, a weekly full backup, with daily incrementals).

The primary advantage gained by using incremental backups is that the incremental backups run more quickly than full backups. The primary disadvantage to incremental backups is that restoring any given file may mean going through one or more incremental backups until the file is found. When restoring a complete file system, it is necessary to restore the last full backup and every subsequent incremental backup.

In an attempt to alleviate the need to go through every incremental backup, a slightly different approach was implemented. This is known as the *differential backup*.

8.2.4.3. Differential Backups

Differential backups are similar to incremental backups in that both backup only modified files. However, differential backups are *cumulative* — in other words, with a differential backup, if a file is modified and backed up on Tuesday night, it will also be backed up on Wednesday night (even if it has not been modified since).

Of course, all newly-modified files will be backed up as well.

Like the backup strategy used with incremental backups, differential backups normally follow the same approach: a single periodic full backup followed by more frequent differential backups.

The affect of using differential backups in this way is that the differential backups tend to grow a bit over time (assuming different files are modified over the time between full backups). However, the benefit to differential backups comes at restoration time — at most, the latest full backup and the latest differential backup will need to be restored.

8.2.5. Backup Media

We have been very careful to use the term "backup media" throughout the previous sections. There is a reason for that. Most experienced system administrators usually think about backups in terms of reading and writing tapes, but today there are other options.

At one time, tape devices were the only removable media devices that could reasonably be used for backup purposes. However, this has changed. In the following sections we will look at the most popular backup media, and review their advantages as well as their disadvantages.

8.2.5.1. Tape

Tape was the first widely-used removable data storage medium. It has the benefits of low media cost, and reasonably-good storage capacity. However, tape has some disadvantages — it is subject to wear, and data access on tape is sequential in nature.

These factors mean that it is necessary to keep track of tape usage (retiring tapes once they have reached the end of their useful life), and that searching for a specific file on tape can be a lengthy proposition.

On the other hand, tape is one of the most inexpensive mass storage media available, and it has a long history of reliability. This means that building a good-sized tape library need not consume a large part of your budget, and you can count on it being usable now and in the future.

8.2.5.2. Disk

In years past, disk drives would never have been used as a backup medium. However, storage prices have dropped to the point where, in some cases, using disk drives for backup storage does make sense.

The primary reason for using disk drives as a backup medium would be speed. There is no faster mass storage medium available. Speed can be a critical factor when your data center's backup window is short, and the amount of data to be backed up is large.

But disk storage is not the ideal backup medium, for a number of reasons:

- Disk drives are not normally removable. One key factor to an effective backup strategy is to get the backups out of your data center and into off-site storage of some sort. A backup of your production database sitting on a disk drive two feet away from the database itself is not a backup; it is a copy. And copies are not very useful should the data center and its contents (including your copies) be damaged or destroyed by some unfortunate set of circumstances.

- Disk drives are expensive (at least compared to other backup media). There may be circumstances where money truly is no object, but in all other circumstances, the expenses associated with using disk drives for backup mean that the number of backup copies will be kept low to keep the overall cost of backups low. Fewer backup copies mean less redundancy should a backup not be readable for some reason.
- Disk drives are fragile. Even if you spend the extra money for removable disk drives, their fragility can be a problem. If you drop a disk drive, you have lost your backup. It is possible to purchase specialized cases that can reduce (but not entirely eliminate) this hazard, but that makes an already-expensive proposition even more so.
- Disk drives are not archival media. Even assuming you are able to overcome all the other problems associated with performing backups onto disk drives, you should consider the following. Most organizations have various legal requirements for keeping records available for certain lengths of time. The chance of getting usable data from a 20-year-old tape is much greater than the chance of getting usable data from a 20-year-old disk drive. For instance, would you still have the hardware necessary to connect it to your system? Another thing to consider is that a disk drive is much more complex than a tape cartridge. When a 20-year-old motor spins a 20-year-old disk platter, causing 20-year-old read/write heads to fly over the platter surface, what are the chances that all these components will work flawlessly after sitting idle for 20 years?


Note

Some data centers back up to disk drives and then, when the backups have been completed, the backups are written out to tape for archival purposes. In many respects this is similar to how AMANDA handles backups.

All this said, there are still some instances where backing up to disk drives might make sense. In the next section we will see how they can be combined with a network to form a viable backup solution.

8.2.5.3. Network

By itself, a network cannot act as backup media. But combined with mass storage technologies, it can serve quite well. For instance, by combining a high-speed network link to a remote data center containing large amounts of disk storage, suddenly the disadvantages about backing up to disks mentioned earlier are no longer disadvantages.

By backing up over the network, the disk drives are already off-site, so there is no need for transporting fragile disk drives anywhere. With enough network bandwidth, the speed advantage you can get from disk drives is maintained.

However, this approach still does nothing to address the matter of archival storage (though the same "spin off to tape after the backup" approach mentioned earlier can be used). In addition, the costs of a remote data center with a high-speed link to the main data center make this solution extremely expensive. But for the types of organizations that need the kind of features this solution can provide, it is a cost they will gladly pay.

8.2.6. Storage of Backups

Once the backups are complete, what happens then? The obvious answer is that the backups must be stored. However, what is not so obvious is exactly what should be stored — and where.

To answer these questions, we must first consider under what circumstances the backups will be used. There are three main situations:

1. Small, ad-hoc restoration requests from users
2. Massive restorations to recover from a disaster
3. Archival storage unlikely to ever be used again

Unfortunately, there are irreconcilable differences between numbers 1 and 2. When a user accidentally deletes a file, they would like it back immediately. This implies that the backup media is no more than a few steps away from the system to which the data is to be restored.

In the case of a disaster that necessitates a complete restoration of one or more computers in your data center, if the disaster was physical in nature, whatever it was that destroyed your computers would also destroy the backups sitting a few steps away from the computers. This would be a very bad state of affairs.

Archival storage is less controversial; since the chances that it will ever be used for any purpose are rather low, if the backup media was located miles away from the data center there would be no real problem.

The approaches taken to resolve these differences vary according to the needs of the organization involved. One possible approach is to store several days worth of backups on-site; these backups are then taken to more secure off-site storage when newer daily backups are created.

Another approach would be to maintain two different pools of media:

- A data center pool used strictly for ad-hoc restoration requests
- An off-site pool used for off-site storage and disaster recovery

Of course, having two pools implies the need to run all backups twice, or to make a copy of the backups. This can be done, but double backups can take too long, and copying requires multiple backup drives to process the copies (and a probably-dedicated system to actually perform the copy).

The challenge for a system administrator is to strike a balance that adequately meets everyone's needs, while ensuring that the backups are available for the worst of situations.

8.2.7. Restoration Issues

While backups are a daily occurrence, restorations are normally a less frequent event. However, restorations are inevitable; they will be necessary, so it is best to be prepared.

The important thing to do is to look at the various restoration scenarios detailed throughout this section, and determine ways to test your ability to actually carry them out. And keep in mind that the hardest one to test is the most critical one.

8.2.7.1. Restoring From the Bare Metal

The phrase "restoring from the bare metal" is system administrator's way of describing the process of restoring a complete system backup onto a computer with absolutely no data of any kind on it — no operating system, no applications, nothing.

Although some computers have the ability to create bootable backup tapes, and to actually boot from them to start the restoration process, the PC architecture used in most systems running Red Hat Linux do not lend themselves to this approach. However, some alternatives are available:

Rescue disks

A rescue disk is usually a bootable CD-ROM that contains enough of a Linux environment to perform the most common system administration tasks. The rescue disk environment contains the necessary utilities to partition and format disk drives, the device drivers necessary to access the backup device, and the software necessary to restore data from the backup media.

Reinstall, followed by restore

> Here the base operating system is installed just as if a brand-new computer were being initially set up. Once the operating system is in place and configured properly, the remaining disk drives can be partitioned and formatted, and the backup restored from the backup media.

Red Hat Linux supports both of these approaches. In order to be prepared, you should try a bare metal restore from time to time (and especially whenever there has been any significant change in the system environment).

8.2.7.2. Testing Backups

Every type of backup should be tested on a periodic basis to make sure that data can be read from it. It is a fact that sometimes backups are performed that are, for one reason or another, unreadable. The unfortunate part in all this is that many times it is not realized until data has been lost and must be restored from backup.

The reasons for this can range from changes in tape drive head alignment, misconfigured backup software, and operator error. No matter what the cause, without periodic testing you cannot be sure that you are actually generating backups from which data can be restored at some later time.

8.3. Disaster Recovery

As a quick thought experiment, the next time you are in your data center, look around, and imagine for a moment that it is gone. And not just the computers. Imagine that the entire building no longer exists. Next, imagine that your job is to get as much of the work that was being done in the data center going in some fashion, some where, as soon as possible. What would you do?

By thinking about this, you have taken the first step of disaster recovery. Disaster recovery is the ability to recover from an event impacting the functioning of your organization's data center as quickly and completely as possible. The type of disaster may vary, but the end goal is always the same.

The steps involved in disaster recovery are numerous and wide-ranging. Here we will present a high-level overview of the process, along with key points to keep in mind.

8.3.1. Creating, Testing, and Implementing a Disaster Recovery Plan

A backup site is vital, but it is still useless without a disaster recovery plan. A disaster recovery plan dictates every facet of the process, including but not limited to:

- What events denote possible disasters, and what people in the organization have the authority to declare a disaster, and thereby put the plan into effect
- The sequence of events necessary to prepare the backup site once a disaster has been declared
- The roles and responsibilities of all key personnel with respect to carrying out the plan
- An inventory of the necessary hardware and software required to restore production
- A schedule listing the personnel that will be staffing the backup site, including a rotation schedule to support ongoing operations without burning out the disaster team members
- The sequence of events necessary to move operations from the backup site to the restored/new data center

Disaster recovery plans often fill multiple looseleaf binders. This level of detail is vital because in the event of an emergency, the plan may well be the only thing left from your previous data center (other than the last off-site backups, of course) to help you rebuild and restore operations.

Such an important document deserves serious thought (and possibly professional assistance to create).

And once such an important document is created, the knowledge it contains must be tested periodically. Testing a disaster recovery plan entails going through the actual steps of the plan: going to the backup site and setting up the temporary data center, running applications remotely, and resuming normal operations after the "disaster" is over. Most tests do not attempt to perform 100% of the tasks in the plan; instead a representative system and application is selected to be relocated to the backup site, and returned to normal operation at the end of the test.

Note

Although it is an overused phrase, a disaster recovery plan must be a living document; as the data center changes, the plan must be updated to reflect those changes. In many ways, an out-of-date disaster recovery plan can be worse than no plan at all, so make it a point to have regular (quarterly, for example) reviews and updates of the plan.

8.3.2. Backup Sites: Cold, Warm, and Hot

One of the most important aspects of disaster recovery is to have a location from which the recovery can take place. This location is known as a *backup site*. In the event of a disaster, a backup site is where your data center will be recreated, and where you will operate from, for the length of the disaster.

There are three different styles of backup sites:

- Cold backup sites
- Warm backup sites
- Hot backup sites

Obviously these terms do not refer to the temperature of the backup site. Instead, they refer to the effort required to begin operations at the backup site in the event of a disaster.

A cold backup site is little more than an appropriately configured space in a building. Everything required to restore service to your users must be procured and delivered to the site before the process of recovery can begin. As you can imagine, the delay going from a cold backup site to full operation can be substantial.

Cold backup sites are the least expensive sites.

A warm backup site is already stocked with hardware representing a reasonable facsimile of that found in your data center. To restore service, the last backups from your off-site storage facility must be delivered, and bare metal restoration completed, before the real work of recovery can begin.

Hot backup sites have a virtual mirror image of your current data center, with all systems configured and waiting only for the last backups of your user data from your off-site storage facility. As you can imagine, a hot backup site can often be brought up to full production in no more than a few hours.

A hot backup site is the most expensive approach to disaster recovery.

Backup sites can come from three different sources:

- Companies specializing in providing disaster recovery services
- Other locations owned and operated by your organization
- A mutual agreement with another organization to share data center facilities in the event of a disaster

Each approach has its good and bad points. For example, contracting with a disaster recovery firm often gives you access to professionals skilled in guiding organizations through the process of creating,

testing, and implementing a disaster recovery plan. As you might imagine, these services do not come without cost.

Using space in another facility owned and operated by your organization can be essentially a zero-cost option, but stocking the backup site and maintaining its readiness is still an expensive proposition.

Crafting an agreement to share data centers with another organization can be extremely inexpensive, but long-term operations under such conditions are usually not possible, as the host's data center must still maintain their normal production, making the situation strained at best.

In the end, the selection of a backup site will be a compromise between cost and your organization's need for the continuation of production.

8.3.3. Hardware and Software Availability

Your disaster recovery plan must include methods of procuring the necessary hardware and software for operations at the backup site. A professionally-managed backup site may already have everything you need (or you may need to arrange the procurement and delivery of specialized materials the site does not have available); on the other hand, a cold backup site means that a reliable source for every single item must be identified. Often manufacturers will work with organizations to craft agreements for the speedy delivery of hardware and/or software in the event of a disaster.

8.3.4. Availability of Backups

When a disaster is declared, it is necessary to notify your off-site storage facility for two reasons:

- To have the last backups brought to the backup site
- To arrange regular backup pickup and dropoff to the backup site (in support of normal backups at the backup site)

Tip

In the event of a disaster, the last backups you have from your old data center are vitally important. Consider having copies made before anything else is done, with the originals going back off-site as soon as possible.

8.3.5. Network Connectivity to The Backup Site

A data center is not of much use if it is totally disconnected from the rest of the organization that it serves. Depending on the disaster recovery plan and the nature of the disaster itself, your user community might be located miles away from the backup site. In these cases, good connectivity is vital to restoring production.

Another kind of connectivity to keep in mind is that of telephone connectivity. You must ensure that there are sufficient telephone lines available to handle all verbal communication with your users. What might have been a simple shout over a cubicle wall may now entail a long-distance telephone conversation; so plan on more telephone connectivity than might at first appear necessary.

8.3.6. Backup Site Staffing

The problem of staffing a backup site is multi-dimensional. One aspect of the problem is determining the staffing necessary to run the backup data center for as long is necessary. While a skeleton crew may be able to keep things going for a short period of time, as the disaster drags on more people will be required to maintain the effort needed to run under the extraordinary circumstances surrounding a disaster.

This includes ensuring that personnel have sufficient time off to unwind, and possibly travel back to their homes. If the disaster was wide-ranging enough to affect peoples' homes and families, additional time must be allotted to allow them to manage their own disaster recovery. Temporary lodging near the backup site will be necessary, along with the transportation required to get people to and from the backup site and their lodgings.

Often a disaster recovery plan includes on-site representative staff from all parts of the organization's user community. This depends on the ability of your organization to operate with a remote data center. If user representatives must work at the backup site, similar accommodations must be made available for them, as well.

8.3.7. Moving Back Toward Normalcy

Eventually, all disasters end. The disaster recovery plan must address this phase as well. The new data center must be outfitted with all the necessary hardware and software; while this phase often does not have the time-critical nature of the preparations made when the disaster was initially declared, backup sites cost money every day they are in use, so economic concerns dictate that the switchover go as quickly as possible.

The last backups from the backup site must be made and delivered to the new data center. After they are restored onto the new hardware, production can be switched over to the new data center.

At this point the backup data center can be decommissioned, with the disposition of all temporary hardware dictated by the final section of the plan. Finally, a review of the plan's effectiveness is held, with any changes recommended by the reviewing committee integrated into the plan.

Index

Symbols

A

B

C

D

E

F

G

H

I

L

M

N

P

Q

R

S

T

U

V

W

Colophon

The Official Red Hat Linux manuals are written in DocBook SGML v4.1 format. The HTML and PDF formats are produced using custom DSSSL stylesheets and custom jade wrapper scripts.

Marianne Pecci <goddess@ipass.net> created the admonition graphics (note, tip, important, caution, and warning). They may be redistributed with written permission from Marianne Pecci and Red Hat, Inc..

The Red Hat Linux Product Documentation Team consists of the following people:

Sandra A. Moore — Primary Writer/Maintainer of the *Official Red Hat Linux x86 Installation Guide*; Contributing Writer to the *Official Red Hat Linux Getting Started Guide*

Tammy Fox — Primary Writer/Maintainer of the *Official Red Hat Linux Customization Guide*; Contributing Writer to the *Official Red Hat Linux Getting Started Guide*; Writer/Maintainer of custom DocBook stylesheets and scripts

Edward C. Bailey — Primary Writer/Maintainer of the *Official Red Hat Linux System Administration Primer*; Contributing Writer to the *Official Red Hat Linux x86 Installation Guide*

Johnray Fuller — Primary Writer/Maintainer of the *Official Red Hat Linux Reference Guide*; Co-writer/Co-maintainer of the *Official Red Hat Linux Security Guide*; Contributing Writer to the *Official Red Hat Linux System Administration Primer*

John Ha — Primary Writer/Maintainer to the *Official Red Hat Linux Getting Started Guide*; Co-writer/Co-maintainer of the *Official Red Hat Linux Security Guide*; Contributing Writer to the *Official Red Hat Linux System Administration Primer*